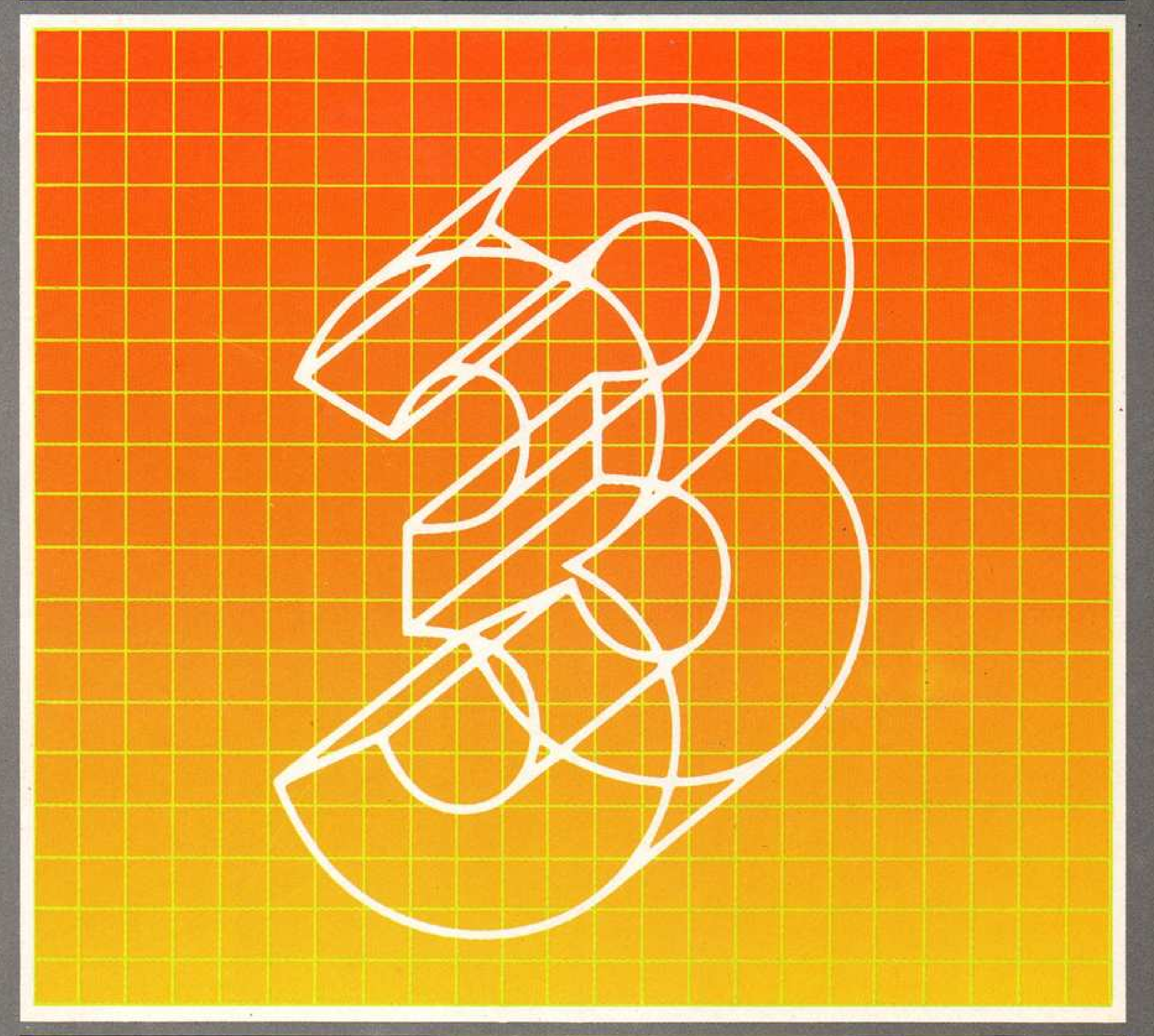
BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



DISFRUTA TU ORDENADOR,
APRENDE Y HAZ COSAS CON EL

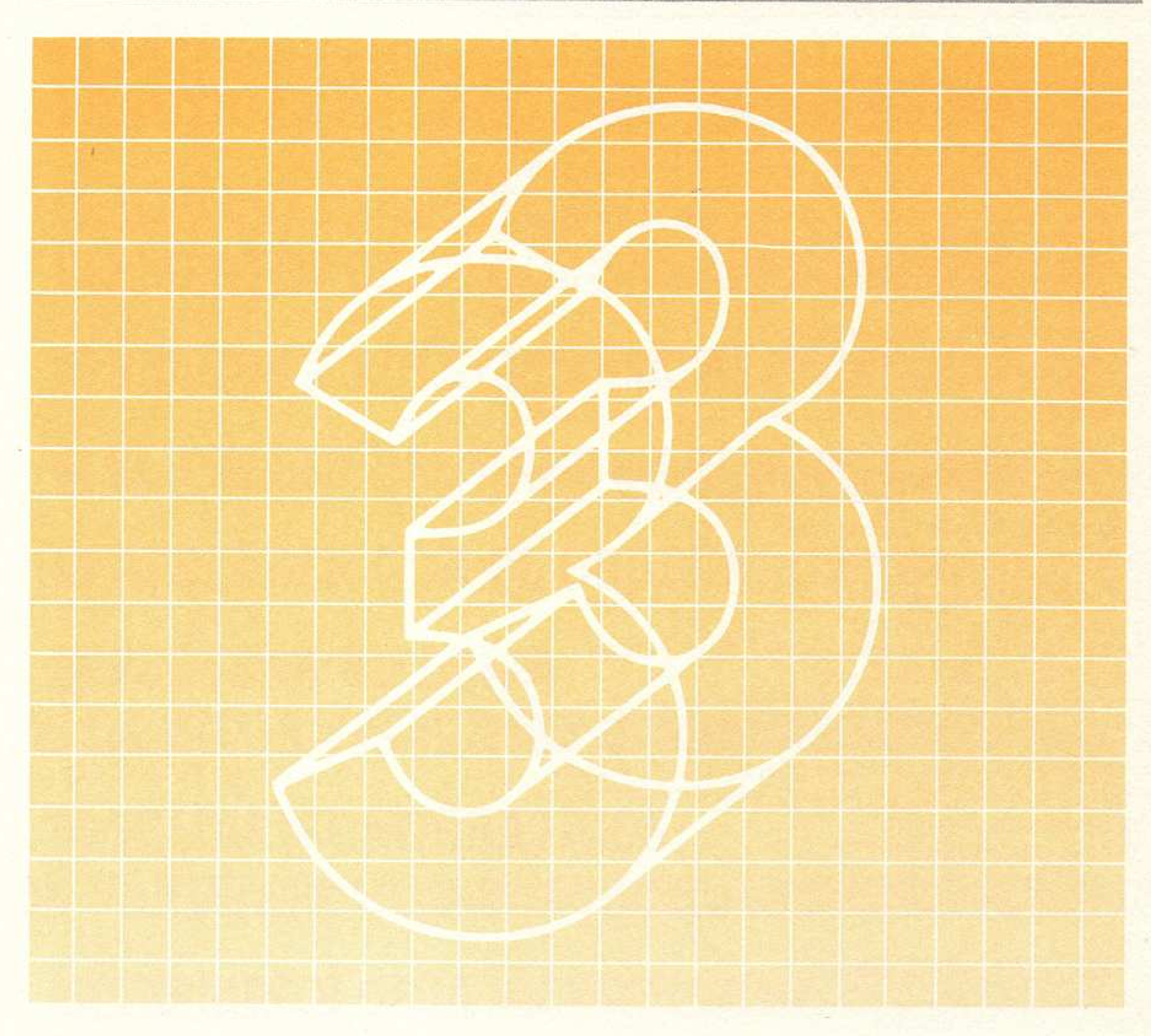
EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD, IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Director de la colecciöin:
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:
BRAVO-LOFISH.

Maquetación:
D. SIMON

Dibujos:
JOSE OCHOA

Tomo 3. «Experiencias prácticas en logo», Carlos Albert, Técnico de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprender con el ordenador», AULA DE INFORMATICA APLICADA: Fernando Suero. Diplomado en Telecomunicación; Alejandro Marcos, Licenciado en Química; Francisco Blanca, Diplomado en Telecomunicación; María José Hernando, Diplomada en Informática; Soledad Tamariz, Diplomada en Telecomunicación. «El Taller del Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:
Sor Angela de la Cruz, 24-7.° G. Teléf. 279 40 36. 28020 Madrid.

Publicidad:
Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:
COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

ISBN del tomo: 84-7688-050-2
ISBN de la obra: 84-7688-047-2

Fotocomposición:
ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:
MATEU CROMO. PINTO (Madrid).

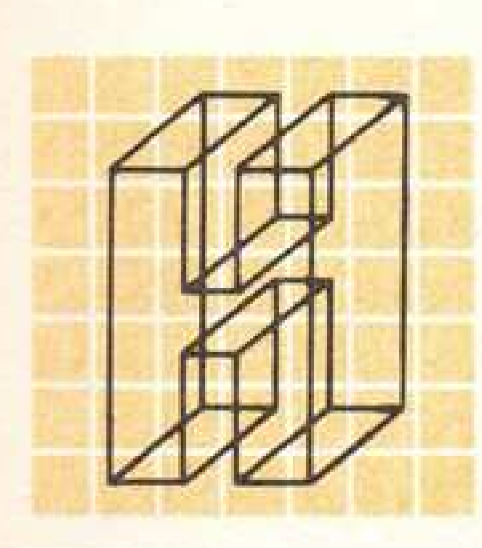
Depósito legal: M-43.185-1986
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S. A.
Sor Angela de la Cruz, 24-7.° G. Teléf. 279 40 36. 28020 Madrid.

Enero, 1987.

INDICE

EXPERIENCIA Y PRACTICAS EN LOGO



EMOS utilizado en numerosas ocasiones una de las órdenes más potentes de las que posee el Logo. Se trata de la orden REPITE. Con ella conseguimos evitar el tener que teclear las mismas cosas un número determinado de veces. Vamos a investigar un poco más sobre ella.

Con la orden REPITE hemos conseguido hacer cuadrados, triángulos, circunferencias y numerosas figuras geométricas más. En todas ellas lo que hacemos es repetir «n» veces la misma orden o grupos de órdenes.

Al igual que ocurre con el resto de las órdenes que existen en el Logo, REPITE tiene una sintaxis, la cual hay que respetar y cumplir obligatoriamente. En el caso de Repite es la siguiente:

REPITE n [Lista de órdenes]

Observa que hay que dejar un espacio entre la palabra REPITE y «n» y lo que se va a repetir tiene que ir encerrado entre corchetes.

Hay versiones de Logo que permiten el omitir el corchete final y, o bien lo pone automáticamente, o bien lo acepta como si existiese. Tampoco es necesario en algunas versiones el tener que dejar un espacio entre «n» y el primer corchete.

Por tanto, la orden REPITE podría también escribirse de la siguiente manera y sería igualmente válida:

REPITE n[Lista de órdenes]

Si el valor de «n» es negativo, se produce un error.

Si el valor de «n» es decimal, éste es redondeado.

El valor de «n» puede oscilar desde valores como 0 hasta valores tan elevados como 32767.

Por ejemplo, si tecleas la orden:

REPITE 200 [REPITE 4 [AV 40 GD 90]]

Obtenemos 200 cuadrados pintados uno encima del otro.

La lista de órdenes que tiene que acompañar a REPITE puede ser tan larga como se quiera, pero por lo menos tiene que contener una orden.

Puede constituir esta lista cualquiera de las órdenes que se utilizan en el Logo. Por tanto, es lógico pensar que dentro de una orden REPITE puede ir perfectamente otra orden REPITE. Su estructura quedaría de la siguiente forma:

REPITE n [REPITE n [lista de órdenes]
lista de órdenes]

Las diferentes órdenes REPITE se van ejecutando de dentro para fuera, es decir, del REPITE más interno al más externo.

Si, por ejemplo, quisiéramos realizar el siguiente dibujo:

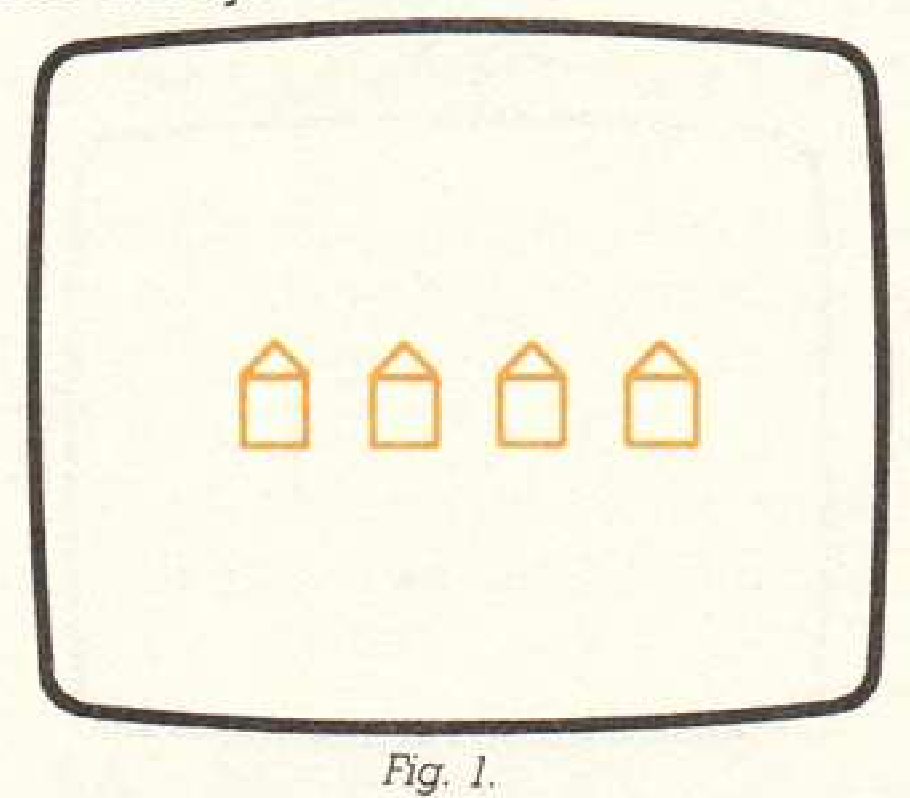


Fig. 1.

Con el Logo puedo dibujar en colores.

Lo único que hay que hacer es dar las órdenes adecuadas para que pinte una sola vez e ir variando la situación de comienzo de donde queremos que nos lo pinte de nuevo.

Las órdenes que lo realizan son las siguientes:

INICIALIZACION

? PM
? BP
? SL
? OT

CENTRANDO DIBUJO

? GI 90
? AV 60
? BL

DIBUJANDO

? REPITE 4 [REPITE 4 [AV 40 GI 90] REPITE 2 [GD 60 AV 40 GI 180] GD 60 SL AV 100 GI 180 BL]

Una vez que inicializamos el estado de la pantalla y de la Tortuga y centramos el dibujo, damos las órdenes que dibujan la figura que se tiene que repetir 4 veces, pero sólo hace falta teclearlas una vez. Cuando la dibuja la primera vez, le ordenamos que se sitúe en donde queremos que lo dibuje la segunda, y así sucesivamente. El REPITE se encarga de hacerlo las veces que deseemos.

Observa estos dos dibujos:

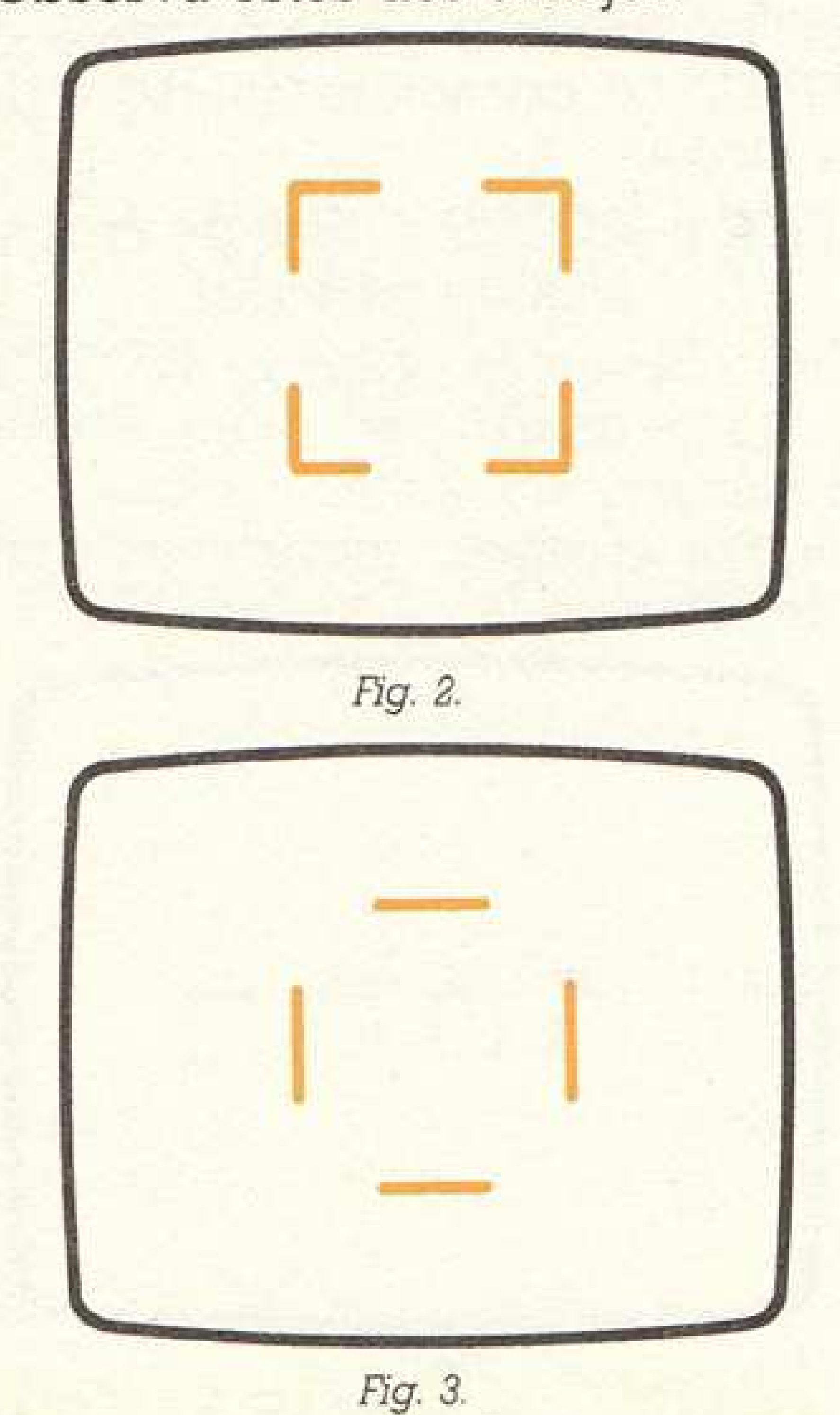


Fig. 2.

Fig. 3.

Date cuenta que el segundo es la parte del dibujo de un cuadrado que le falta al primero.

Teclea e introduce las siguientes órdenes:

INICIALIZANDO

? PM
? SL
? BP
? OT

PRIMER DIBUJO

? REPITE 4 [BL AV 20 SL AV 20 BL AV 20 GD 90]

SEGUNDO DIBUJO

? IL
REPITE 4 [AV 60 GD 90]

En el primer dibujo realizamos un cuadrado, pero dejamos en cada lado un espacio sin pintar.

En el segundo, realizamos un cuadrado completo y de igual tamaño que el anterior y sobre él.

¿Qué ha ocurrido? Al dar la orden IL, en el segundo dibujo, ha empezado a dibujarse el cuadrado, pero al pasar por encima de algo que está pintado, lo borra, y pinta donde se encuentra que no hay nada dibujado.

En definitiva, lo que ocurre es que se invierte el dibujo. Esto se consigue con la orden IL (Invierte lápiz).

Cambia en las órdenes anteriores el último bloque por:

SEGUNDO DIBUJO

? IL
? REPITE 20 [REPITE 4 [AV 60 GD 90]]

Hemos obtenido un curioso efecto. Se crea como un circuito cerrado de los lados girando hacia una misma dirección.

El teclado

Dependiendo del ordenador que se posea, existen un cierto número de teclas que realizan una función determinada. En este caso vamos a ver algunas de las teclas que tienen funciones específicas dentro del Logo y que nos son de gran ayuda a la hora de trabajar.

Las órdenes Logo se llaman primitivas.

Si posees un ordenador MSX:

• Las teclas de función F1-6, F2-7 y F3-8 nos permiten pasar a los modos de pantalla, Texto, Mixta y Gráficos, respectivamente, con sólo presionarlas. Así nos evitamos el tener que teclear la orden correspondiente para pasar a cada modo de pantalla.

• Si presionamos a la vez las teclas CTRL (control) e Y, obtendremos la última línea que hemos escrito.

• Si presionamos a la vez las teclas CTRL y STOP, interrumpimos la ejecución de lo que se esté realizando en ese momento.

Si posees un SPECTRUM:

• Si pulsas las teclas CAPS y BREAK/SPACE simultáneamente, obtenemos una interrupción.

• Con E MODE R obtenemos la última línea memorizada a partir de la posición del cursor.

• Con SYS Y y SYS V obtenemos, respectivamente, los corchetes, [y].

Si posees un compatible PC:

• Si presionas simultáneamente CTRL y NUMLOCK, se interrumpe la ejecución, pudiéndose reemprender pulsando cualquier otra tecla.

• Las teclas F3 y F4 marcan el principio y el final de un bloque.

• La tecla F5 copia el bloque marcado anteriormente.

• Si presionamos simultáneamente CTRL, ALT y [, obtenemos el corchete abierto.

• Si presionamos simultáneamente CTRL, ALT y], obtenemos el corchete cerrado.

¡¡ATENCION!! En algunos PC hay que escribir las órdenes con letras mayúsculas.

Las próximas órdenes nos dibujan un tablero de ajedrez:

INICIALIZANDO

? PM
? SL
? BP
? OT

CENTRANDO DIBUJO

? GI 90 AV 64
? GD 90 RE 50
? BL
? PONCL 1

DIBUJO TABLERO

? REPITE 4 [AV 128 GD 90] GD 90

CUADRICULADO DEL TABLERO

? REPITE 2 [REPITE 8 [AV 16 GI 90 AV 128 RE 128 GD 90] GI; 90

COLOCACION
PARA PINTAR CUADRADOS

? GD 90 SL RE 8 GI 90 AV 8

PINTO CUADRADOS NEGROS

? REPITE '4 [REPITE 4 [BL RELLENA SL AV 32] RE 160 GI 90 AV 32 GD 90 AV 32] GD 90 AV 16 GI 90 AV 16
? REPITE 4 [REPITE 4 [BL RELLENA SL AV 32] RE 160 GD 90 AV 32 GI 90 AV 32]

PINTO CUADRADOS BLANCOS

? PONFONDO 15
? PONCL 4
? RELLENA

Bueno, ya tienes lo principal para jugar al ajedrez. Vamos a explicar un poco cómo lo hemos realizado.

Lo primero que hacemos, como en todos los demás programas, es inicializar la pantalla como el estado de la Tortuga. Luego centramos el dibujo.

Comenzamos a dibujar; un simple cuadrado me determina el tamaño del tablero. A continuación cuadriculamos el tablero para obtener los casilleros de un tablero de ajedrez. Esto lo conseguimos trazando 8 líneas horizontales y 8 verticales. Ya sólo nos queda pintar los cuadrados que van en negro y los que van en blanco.

Los negros los dibujamos rellenando casilleros salteados del color que hemos establecido con la orden PONCL, en este caso el negro, ya que hemos dado valor 1 en la orden PONCL.

Los blancos los obtenemos simplemente dando un color a todo el fondo. En este caso el blanco. Lo realizamos con la orden PONFONDO 15.

No te preocupes si no entiendes las nuevas órdenes que hemos utilizado en este

La orden «REPITE» me evita tener que teclear la misma orden un número determinado de veces.

dibujo. Ya las estudiaremos con más detenimiento.

Veamos ahora alguna de las órdenes que ya utilizamos con bastante frecuencia. Son las órdenes con las que podemos decir a la Tortuga que pinte o que no pinte.

Con la orden BAJALAPIZ (BL) activamos el lápiz de la Tortuga, y todo desplazamiento que realice desde este momento lo hará dejando un rastro tras de sí. El lápiz estará activado hasta que le demos la orden oportuna para desactivarlo, y no dejará de pintar en sus desplazamientos hasta dicho momento.

La orden que desactiva el lápiz es SUBELAPIZ (SL). Desde el momento en que se dé esta orden, todo desplazamiento de la Tortuga lo hará sin dejar un rastro tras de sí.

El lápiz estará desactivado hasta que lo volvamos a activar, lo que conseguimos con la orden BL.

Aquí tienes un curioso ejemplo de cómo poder tener más de una Tortuga a la vez y conseguir que cada una de ellas vaya por su lado.

INICIALIZANDO

```
? PM
? SL
? BP
? OT
```

DETERMINO EL NUMERO DE TORTUGAS

```
? ACTIVA [0 1 2 3]
```

COLOCO CADA UNA EN UN LUGAR

```
? ACTIVA [0] GI 45 AV 80
? ACTIVA [1] GD 45 AV 80
? ACTIVA [2] GI 135 AV 40
? ACTIVA [3] GD 135 AV 40
```

MUEVO LAS CUATRO

```
? ACTIVA [0 1 2 3] MT
? REPITE 100 [GD 10 AV 5 GD 5 AV 5 GI 3 AV 5]
```

Determino el número de tortugas con la orden ACTIVA, en este caso cuatro, la 0, la 1, la 2 y la 3.

Luego las coloco cada una en un punto determinado de la pantalla. Una vez dispersadas las hacemos visibles y les ordenamos que empiecen a moverse.

Si quieres observar la trayectoria de sus movimientos cambia la orden SL por BL.

Cómo cargar correctamente el Logo

Si posees un ordenador MSX:

— Con el ordenador apagado, introduce el cartucho de Logo en la ranura que tiene destinado para dicho fin.

— Enciende el ordenador.

— El Logo se cargará automáticamente y se autoejecutará.

Si posees un SPECTRUM:

— Coloca la cinta cassette en la lectora-grabadora de cintas.

— Rebobina la cinta hasta el principio.

— Teclea: LOAD "" e introdúcelo.

— Conecta la tacla PLAY de la lectora-grabadora.

— El Logo se empezará a cargar y se ejecutará automáticamente cuando termine de cargarse.

— Si ves que no se carga, ajusta el volumen y repite la operación anterior.

Si posees un compatible PC:

— Introduce el disqquete del Sistema Operativo en la unidad de disquete A.

— Carga el Sistema Operativo de la forma que lo explica el Manual del DOS de tu ordenador.

— Una vez cargado el Sistema Operativo, introduce el disquete del LOGO en la unidad A.

— Consulta en el Manual que acompaña a tu versión del Logo el nombre que hay que poner a la hora de cargarlo y escribe:

A > (nombre)

— EL Logo se cargará y se ejecutará automáticamente.

Con la orden «RELLENA» consigo llenar un espacio en un tiempo mínimo.

Cuadro Resumen

INVIERTELAPIZ (IL)

Deja el lápiz de la Tortuga en un estado en el cual si pasa por encima de algo dibujado lo borra y dibuja si no hay nada dibujado por donde pasa.

PONCL n

Pone el color que determina n al lápiz de la Tortuga. El valor de n puede oscilar desde 0 a 15.

PONFONDO n

Pone del color que determina n, al fondo de toda la pantalla. El valor de n puede oscilar desde 0 a 15.

RELLENA

Llena del color en que se encuentre el lápiz de la Tortuga un espacio cerrado, el cual contiene a la Tortuga en su interior.

ACTIVA [n1 n2 n3...]

Determina el número de Tortugas que se quiere tener. Cada valor de n activará a una Tortuga en concreto, que será reconocida por dicho valor de n.

Ejercicios

1. *Cómo abreviarías las siguientes órdenes:*
   - GD 90 AV 20 GD 90 AV 20 GD 90 AV 20
   - AV 10 GD 45 AV 10 GD 45 GI 90 RE 60 AV 10 GD 45 AV 10 GD 45
   - REPITE 4 [AV 20 GI 20 GI 20]

2. *Dibuja con la orden REPITE el siguiente dibujo:*

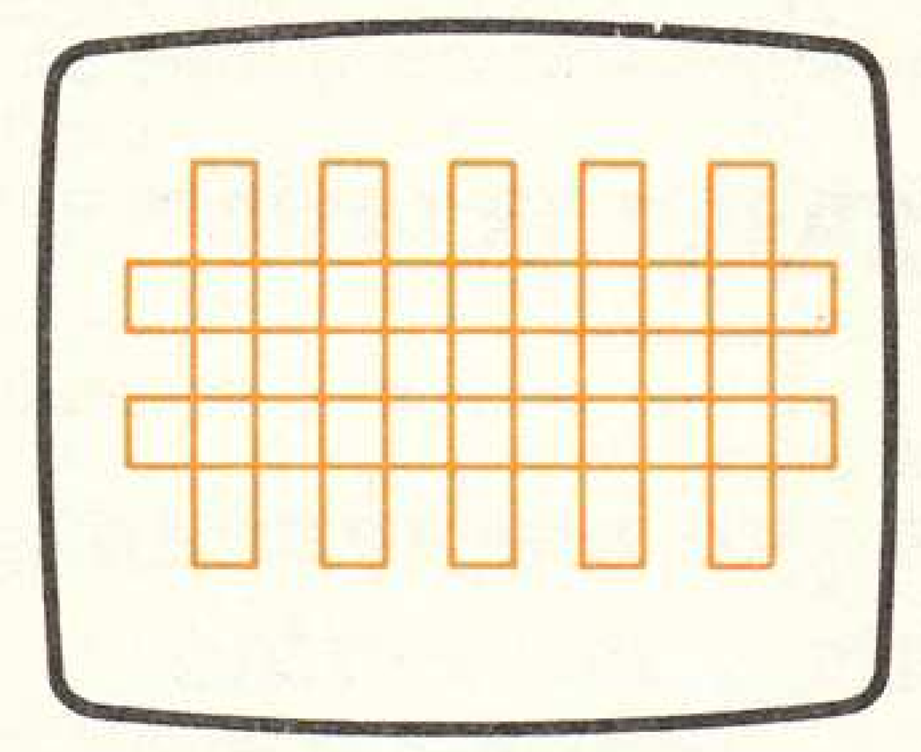


Fig. 4.

3. *¿Qué dibujo se obtiene si invertimos el lápiz de la Tortuga sobre este dibujo?*

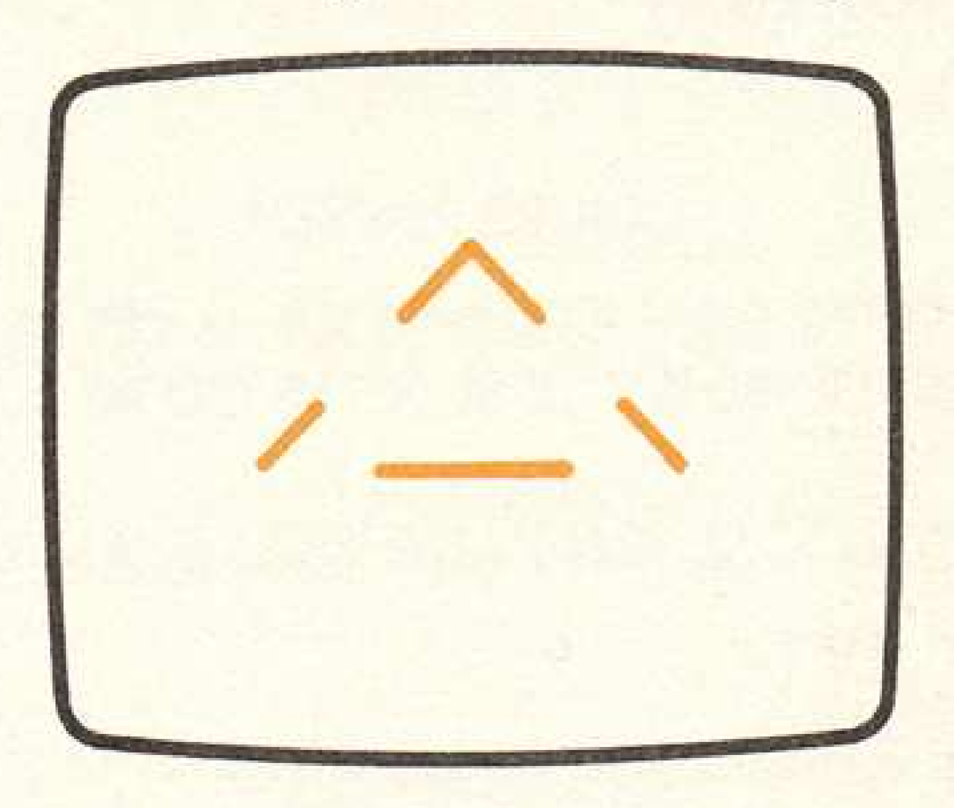


Fig. 5.

4. *Haz el siguiente dibujo:*

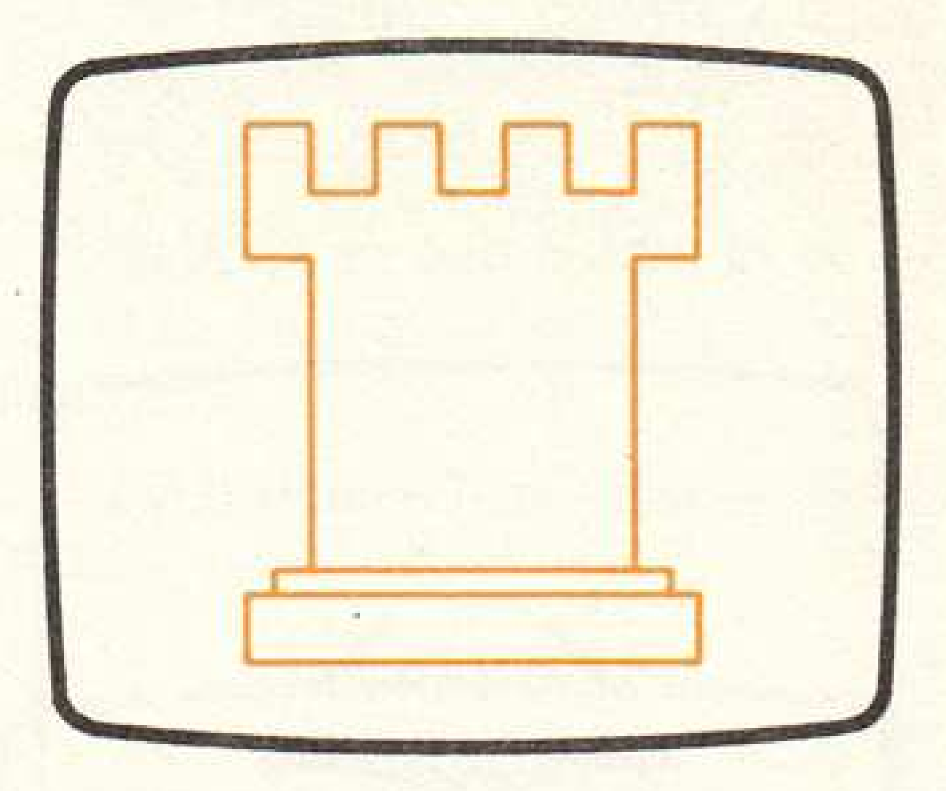


Fig. 6.

5. *¿Son correctas las siguientes órdenes?*
   - REPITE 399 [RE 500]
   - IL 5
   - REPITE 10 [AV 35]
   - SUBELAPIZ AV 2 IL
   - REPITE 20 [AV 4 GD 10]

Solución a los ejercicios

1.

- REPITE 3[GD 90 AV 20]
- REPITE 2[REPITE 2[AV 10 GD 45] GI 90 RE 60]
- REPITE 4[AV 20 GI 40]

2. INICIALIZACION

? PM
? SL
? BP
? OT

En Logo voy a poder crear mis propias órdenes.

CENTRANDO DIBUJO

? GI 90 AV 90
? GD 90 RE 20
? BL

DIBUJO POSTES

? REPITE 5[AV 60 GD 90 AV 15 GD 90 AV 60 GD 90 AV 15 SL RE 35 GD 90 BL]

COLOCACION PARA TRAVESAÑOS

? SL AV 10
? GI 90
? BL

DIBUJO TRAVESAÑOS

? REPITE 2[AV 190 GD 90 AV 10 GD 90 AV 190 GD 90 AV 10 SL RE 30 GD 90 BL]

3. Este es el dibujo que se obtiene:

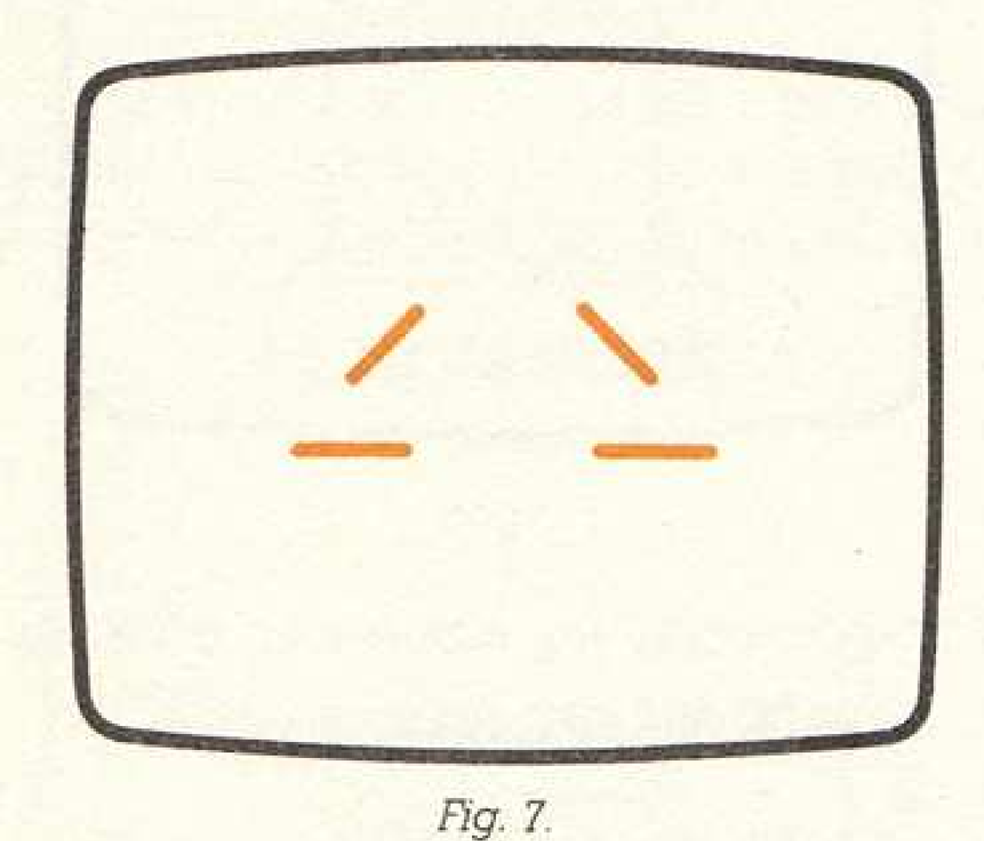


Fig. 7.

4. INICIALIZACION

? PM
? SL
? BP
? MT

CENTRANDO DIBUJO

? GI 90 AV 35
? GD 90 RE 40
? BL

1.ª PEANA

? AV 10 GD 90
? AV 70 GD 90
? AV 10 GD 90
? AV 70

2.ª PEANA

? SL
? RE 5 GD 90 AV 10
? BL
? AV 5 GD 90
? AV 60 GD 90
? AV 5

TORREONES

? RE 5 GD 90 AV 10 GD 90 AV 70
? GD 90 AV 15 GI 90 AV 10
? REPITE 3[REPITE 3 [AV 10 GI 90] RE 10 GI 90]
? REPITE 2[AV 10 GI 90]
? AV 20 GI 90 AV 15
? GD 90 AV 70

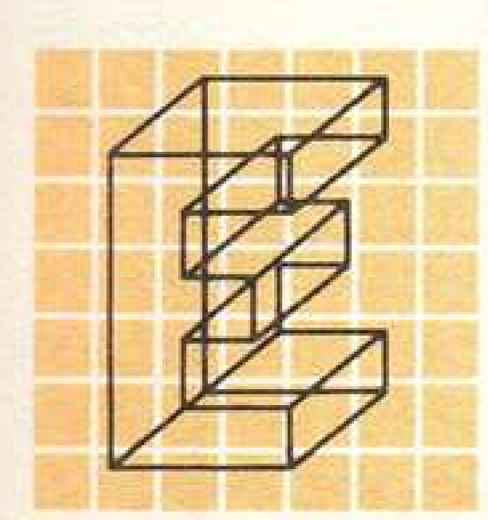
5. REPITE 399[RE 500]: Correcta

- IL 5: Incorrecta, IL no admite ningún valor.
- REPITE 10[AV 35]: Correcta
- SUBELAPIZ AV 2 IL: Correcta
- REPITE-20[AV 4 GD 10]: Incorrecta, REPITE no admite valores negativos

- ***La orden RELLENA es igual a la orden FILL en otros lenguajes.***
- ***Existen diferentes teclas que me ayudan a programar más fácilmente y me evitan tener que perder tiempo.***

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

Los primeros movimientos



L movimiento es una de las características más importantes que tienen todos los juegos comerciales. En la pantalla de nuestro ordenador podemos ver cómo se mueve desde una pequeña pelota hasta un marciano, pasando por un coche o un perro que corre detrás de un gato. Para realizar el movimiento de todos estos personajes se utilizan las mismas técnicas y el mismo principio básico que para la creación de una película de dibujos animados.

Dicho principio básico consiste en dibujar el personaje en una determinada posición y fotograma a fotograma, ir variando la posición muy poquito a poco. Haciendo esto, de una forma repetitiva y continua, se consigue la sensación de que la figura se mueve.

Este mismo principio se utiliza para la creación de animación en los ordenadores. Primero se imprimiría la figura que se va a mover. Se borraría. Se volvería a dibujar en una posición próxima a la anterior. Se volvería a borrar. Se volvería a imprimir y así sucesivamente.

Por supuesto, la figura a mover puede estar compuesta de un solo carácter, una pelota, por ejemplo, o de varios, en el caso de un perro que corre.

Una vez visto esto vamos a ver cómo podemos hacer que un objeto se mueva por la pantalla.

Movimiento de un carácter

Imaginemos que la letra "O" mayúscula es una pequeña pelota y que queremos moverla por la pantalla. Podemos moverla de diferentes formas:

1. De izquierda a derecha, o de derecha a izquierda, en línea recta.
2. De arriba a abajo, o de abajo a arriba, en línea recta.
3. En diagonal con distintos ángulos de inclinación.
4. En círculo o haciendo un movimiento circular.

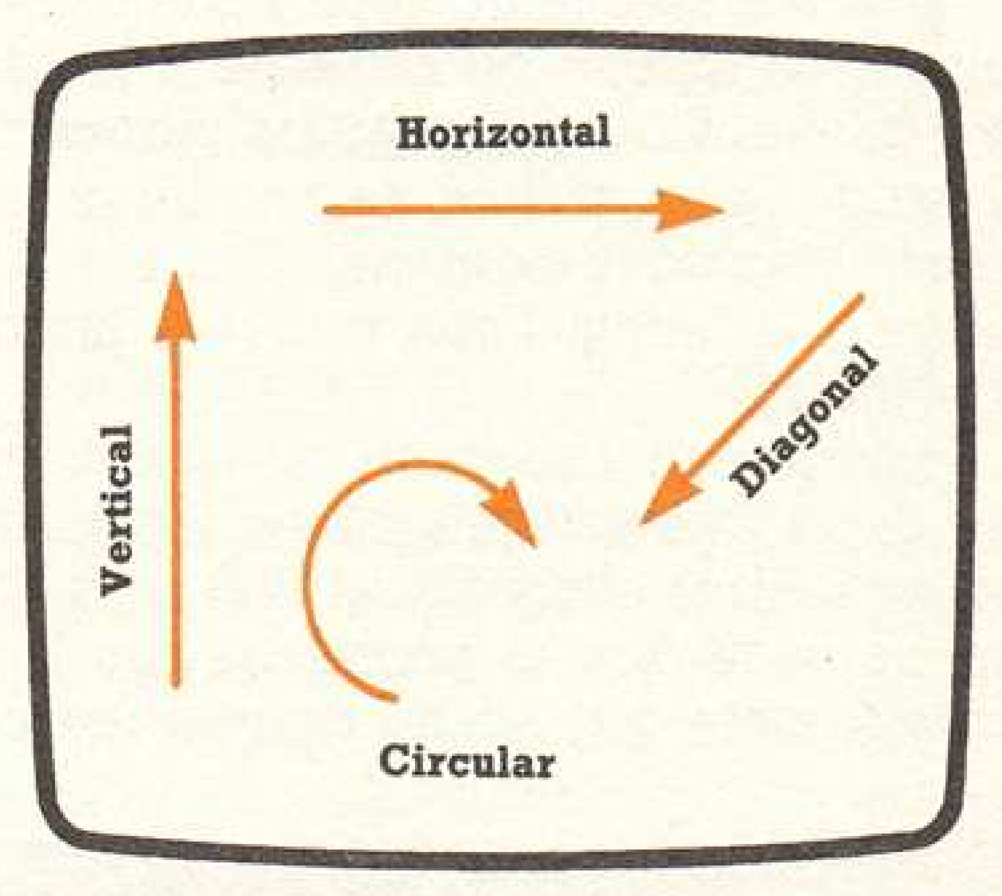


Fig. 1. Tipos de movimientos posibles.

Analicemos todos estos casos uno a uno.

Movimiento de derecha a izquierda en línea recta

Este es el ejemplo más sencillo. Imaginemos que queremos mover un carácter por la pantalla de derecha a izquierda. Este carácter podría ser un punto (.), por ejemplo. Un programa que podríamos utilizar sería el número 1. Introdúcelo en tu ordenador, y ejecútalo, antes de seguir leyendo.

```
 10 REM *************************************
 20 REM * MOVIMIENTO DE DERECHA A IZQUIERDA *
 30 REM * EN EL MSX, IBM Y AMSTRAD          *
 40 REM *************************************
 50 REM
 60 LET Y=10
 70 FOR I=40 TO 1 STEP -1
 80    CLS
 90    LOCATE Y,I:REM <-- EN EL MSX CAMBIARLO POR: LOCATE I,Y
100    PRINT "."
110 NEXT I
120 LET Y=Y+1
130 IF Y=20 THEN LET Y=1
140 GOTO 70
```

Los usuarios del Commodore tendrán que realizar las siguientes modificaciones:

Línea 70 FOR I=39 TO 0 STEP -1
Línea 80 PRINT "<SHIFT-HOME>"
Línea 90 GOSUB 9900

Para que funcione correctamente habrá que unir este programa a la subrutina LOCATE PARA COMMODORE que se dio en el primer tomo.

Para el Spectrum habrá que hacer los siguientes cambios:

Línea 70 FOR I=32 TO 0 STEP -1
Línea 90 PRINT AT I,X;

Una vez ejecutado veréis que el pequeño puntito se mueve de derecha a izquierda en la pantalla. Si os fijáis, podréis comprobar que cuando llega al final de una línea aparece al principio de la siguiente. Cuando el puntito llega a la última línea vuelve a aparecer en la primera.

Aunque el programa funcione, no se puede decir que esté muy bien realizado. El hecho de utilizar el comando CLS para borrar el puntito hace que el programa sea lento y malo. Imagínate que dicho puntito fuese una estrella que se mueve por el cielo. Si el cielo está plagado de estrellas y borramos la pantalla tendremos que volver a dibujar las estrellas cada vez que movemos nuestra estrella fugaz. Por tanto, es mejor borrar sólo la estrella antes de ponerla en la siguiente posición.

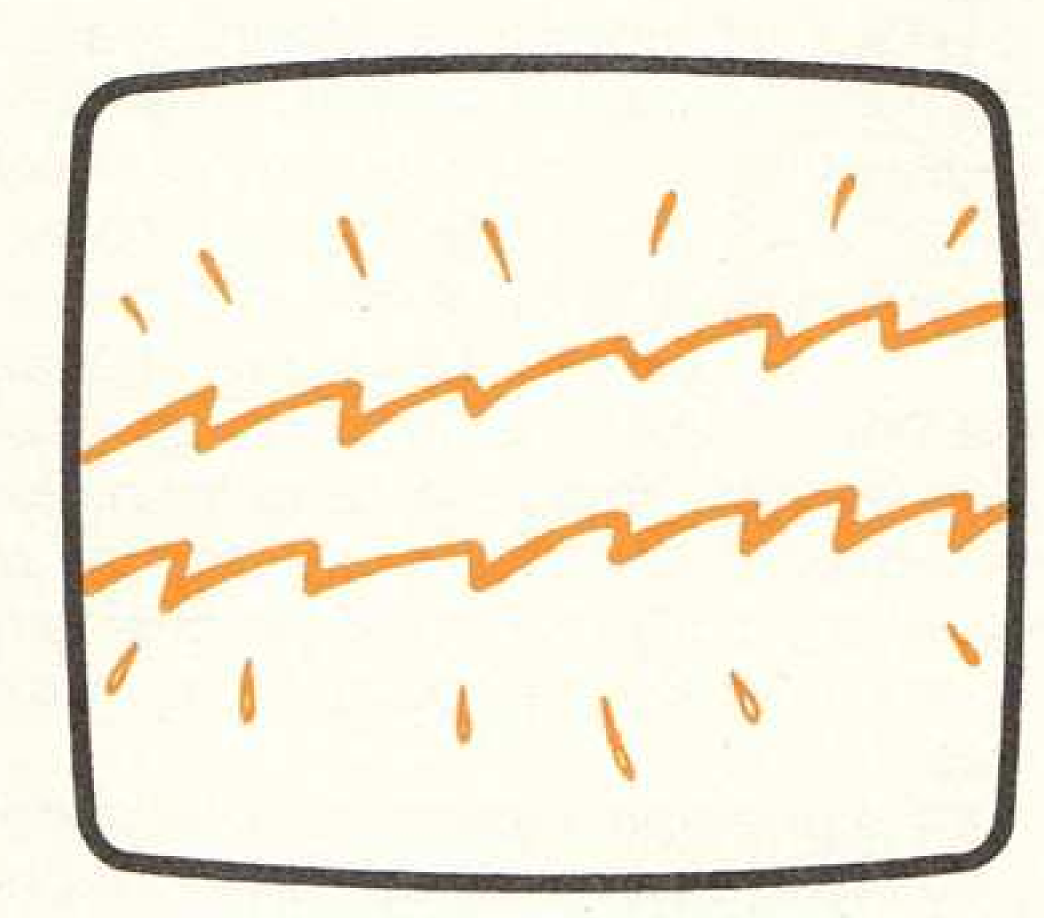


Fig. 2. Si usamos el comando CLS para borrar el punto, la pantalla parpadeará y hará un efecto muy poco agradable.

Si introducimos el programa número 2 veremos que el movimiento es mucho más agradable y suave, y que no se borran las demás estrellas.

```
 10 REM ***********************************************
 20 REM * MOVIMIENTO HORIZONTAL DE UNA ESTRELLA FUGAZ *
 30 REM ***********************************************
 40 REM
 50 REM *** DIBUJO DEL FIRMAMENTO ***
 60 REM
 65 CLS
 70 FOR I=1 TO 50
 80    LET X=INT(39*RND)
 90    LET Y=INT(20*RND)
100    LOCATE Y,X
110    PRINT "."
120 NEXT I
130 REM *** MOVIMIENTO DE LA ESTRELLA FUGAZ ***
140 REM
150 LET Y=10
160 FOR I=39 TO 1 STEP -1
170    LOCATE Y,I+1
180    PRINT " "
190    LOCATE Y,I
200    PRINT "."
```



```
210     FOR J=1 TO 100
220     NEXT J
230 NEXT I
240 REM
250 REM *** DESAPARECE LA ESTRELLA FUGAZ ***
260 REM
270 LOCATE Y,1
280 PRINT " "
290 END
```

Las variaciones que hay que hacerle al programa, dependiendo del ordenador, son las que se dan a continuación:

COMMODORE

Línea 65 PRINT "<SHIFT-HOME>"
Línea 100 GOSUB 9900
Línea 170 LET X=I+1:GOSUB 9900
Línea 190 LET X=I:GOSUB 9900
Línea 270 LET X=I:GOSUB 9900

Para ejecutar este programa, y todos los demás programas de este tomo, es necesario unirlos a la rutina 'LOCATE PARA COMMODORE' que se dio en el tomo 1.

SPECTRUM

Línea 80 LET X=INT(31*RND)
Línea 100 PRINT AT Y,X;
Línea 160 FOR I=1 30 TO 1 STEP -1
Línea 170 PRINT AT Y,I+1;
Línea 190 PRINT AT Y,I;
Línea 270 PRINT AT Y,1;
Línea 290 STOP

MSX

Línea 80 LET X=INT(38*RND(1))
Línea 90 LET Y=INT(20*RND(1))

También hay que cambiar de orden los argumentos de todas las sentencias LOCATE. Si, por ejemplo, aparece en el listado: LOCATE Y,X; nosotros tendremos que poner: LOCATE X,Y.

AMSTRAD

Línea 160 FOR I= 38 TO 1 STEP -1

Para el IBM no hace falta realizar ningún cambio, pero se recomienda que la pantalla del ordenador se encuentre en el modo de 40 columnas por línea (esto también vale para el Amstrad y MSX). Recuerda que esto se hace utilizando la sentencia WIDTH.

Como puedes ver, si ejecutas el programa, la estrella fugaz aparece por la derecha de la pantalla, se mueve hacia la izquierda y desaparece al tocar el borde izquierdo. Si dio la casualidad de que en la línea en la que se mueve la estrella fugaz había otra estrella, ésta desaparecerá. Este programa no mira si, en el camino de la estrella, hay alguna otra para no borrarla. Realizar esto es algo más complicado que lo que hemos visto hasta ahora, pero lo iremos viendo en tomos sucesivos.

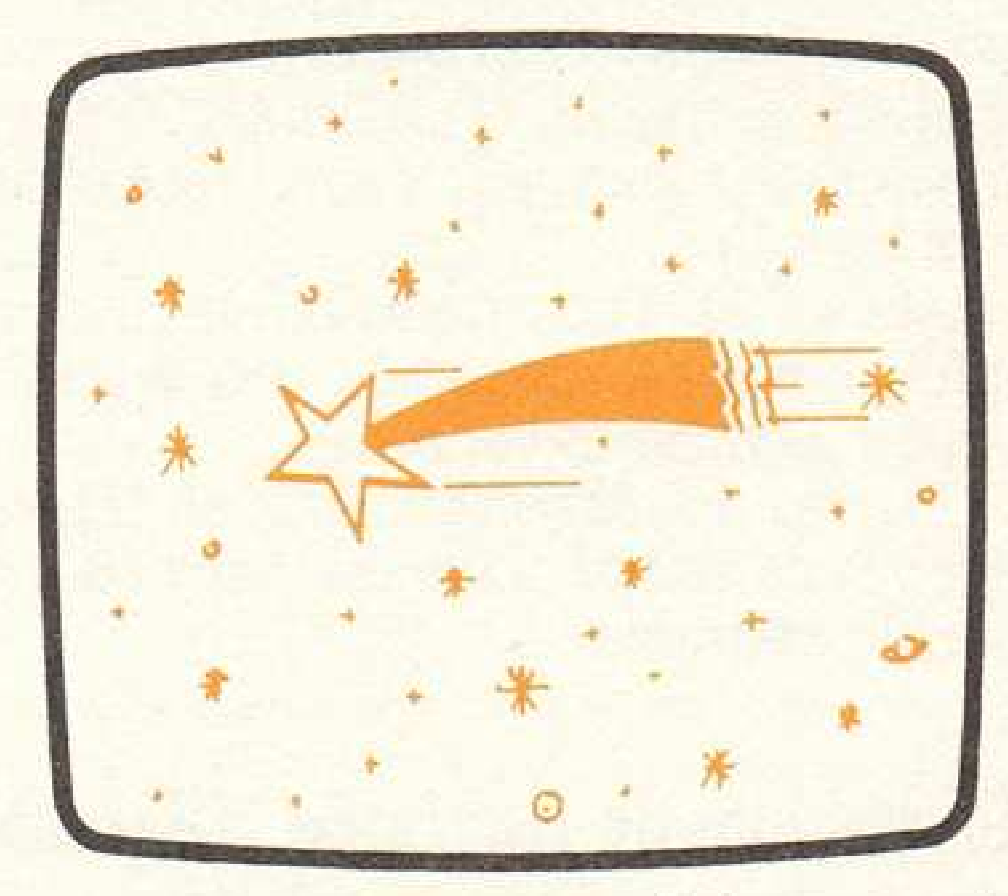


Fig. 3. La forma de dibujar una estrella fugaz, como la de Belén, que se mueva por la pantalla aparecerá en tomos sucesivos.

El funcionamiento del programa es el siguiente:

Entre las líneas 70 y 120 se imprimen 50 estrellas en toda la pantalla de forma aleatoria. Esto se consigue con el uso de la función RND. RND nos devuelve un valor entre 0 y 1. Dicho valor, al multiplicarlo por un cierto número (39 ó 20, según el caso), se convierte en un número entre 0 y 38 (si multiplicamos por 39) o entre 0 y 19 (al multiplicarlo por 20). Cada par de valores es utilizado para imprimir, en una cierta columna de cierta línea, un punto que hará las funciones de una estrella.

Entre las líneas 150 y 220 hacemos que nuestra estrella fugaz se mueva por la pantalla. La forma de hacerlo, analizando línea a línea, es la siguiente:

Línea 150. En esta línea asignamos a la variable Y el valor de la fila por donde se moverá la estrella.

Línea 160. Aquí empieza un bucle que nos va diciendo la columna en la que se colo-

cará la estrella. Como el movimiento es de derecha a izquierda el bucle va de 39 (30 en el caso del Spectrum) hasta 1. Por ello, el incremento (STEP) es negativo y va de -1 en -1.

Líneas 170 y 180. Se borra lo que hubiese en la columna anterior a la que nos encontramos ahora. En la primera vuelta del bucle, cuando I = 39, esto no tiene ningún efecto. Pero, a partir de la segunda vuelta, como la estrella se encuentra impresa en la columna 39 e I es igual a 38, borrará la estrella fugaz.

Líneas 190 y 200. Se imprime en la columna I de la fila Y un punto (.).

Líneas 210 y 220. Se realiza un bucle en vacío que hace de retardo entre impresión e impresión. Si se quiere que la estrella se mueva más deprisa (o más despacio) sólo hay que variar la longitud del bucle (el número 100 que aparece detrás de la palabra TO). Cuanto mayor sea dicha longitud, mayor será el tiempo de retardo.

Línea 230. Termina el bucle principal. Se repite la misma operación una y otra vez hasta que I alcanza el valor de 1.

Una vez que la estrella está en la columna número uno se borra (líneas 270 y 280) y se termina el programa.

Movimiento de izquierda a derecha en línea recta

Variar el programa anterior para que el movimiento sea de izquierda a derecha, en vez de derecha a izquierda, es muy sencillo. Sólo tenemos que variar unas cuantas líneas:

```
Línea 160 FOR I=2 TO 40
Línea 170    LOCATE Y,I-1
Línea 180    PRINT " "
Línea 190    LOCATE Y,I
Línea 200    PRINT "."
Línea 270 LOCATE Y,40
```

Los usuarios del MSX, Commodore y Spectrum tienen que hacer las modificaciones pertinentes tal y como se dieron un poco más arriba.

Estas modificaciones para hacer que el movimiento sea de izquierda a derecha son un poco rupestres y se pueden mejorar con el fin de ahorrar memoria. La forma sería:

```
Línea 160 FOR I=1 TO 39
Línea 170    LOCATE Y,I
Línea 180    PRINT "."
```

borrando las líneas 190 y 200.

La forma de trabajar es muy parecida a la que hemos visto anteriormente, pero, como el incremento va de uno en uno, el valor de I va aumentando en vez de disminuir. Gracias a esto conseguimos que el movimiento sea de izquierda a derecha.

Movimiento de abajo a arriba en línea recta

El movimiento de un carácter de abajo hacia arriba por la pantalla es muy parecido a lo que ya hemos visto. La diferencia entre los movimientos horizontales y los verticales consiste en que, mientras en el horizontal lo que varía es la columna, en el vertical va variando la fila. El programa número 3 nos muestra un ejemplo de esto.

```
 10 REM **************************************
 20 REM * MOVIMIENTO DE UNA 'O' A LO LARGO   *
 30 REM * DE UNA CARRETERA DE ABAJO A ARRIBA *
 40 REM **************************************
 50 REM
 60 CLS
 70 FOR I=20 TO 1 STEP -1
 80    LOCATE I,19
 90    PRINT "!   !"
100 NEXT I
110 REM
120 REM *** MOVIMIENTO ***
130 REM
140 LET X=21
150 FOR I=20 TO 1 STEP -1
160    LOCATE I+1,X
170    PRINT " "
180    LOCATE I,X
190    PRINT "O"
200    FOR J=1 TO 100
210    NEXT J
220 NEXT I
230 REM
240 REM *** LA 'O' DESAPARECE ***
250 REM
260 LOCATE 1,X
270 PRINT " "
280 END
```

En este programa podemos apreciar cómo la letra O se mueve de abajo hacia arriba a lo largo de una especie de carretera.

Este programa puede funcionar sin problemas en el Amstrad e IBM en el modo de 40 columnas por línea. Para los demás ordenadores las modificaciones son las siguientes:

COMMODORE

```
Línea  60 PRINT "<SHIFT-HOME>
Línea  80 LET Y=I:GOSUB 9900
Línea 160 LET Y=I+1:GOSUB 9900
Línea 180 LET Y=1:GOSUB 9900
Línea 260 LET Y=1:GOSUB 9900
```

Como siempre, hay que unir este programa con la rutina LOCATE PARA COMMODORE que se dio en el tomo 1 en la sección de trucos de programación.

MSX

Hay que cambiar de orden los argumentos de la sentencia 'LOCATE'. Según esto, donde ponga: LOCATE I,19; nosotros tendremos que invertir el orden y poner: LOCATE 19,I.

También es conveniente poner la pantalla en 40 columnas por línea mediante la sentencia WIDTH 40. Ni que decir tiene que para esto es necesario que estemos en SCREEN 0.

SPECTRUM

Hay que cambiar todas las sentencias 'LOCATE' por PRINT AT. Por ejemplo, si en la línea 80 pone:

80 LOCATE I,19

nosotros tendremos que cambiarlo por:

80 PRINT AT I,19

sin olvidarnos de poner el punto y coma (;) al final.

El funcionamiento del programa es muy sencillo. Entre las líneas 70 y 100 se dibuja la carretera de abajo hacia arriba. Se usa el signo de admiración (!) para señalar los bordes.

El movimiento propiamente dicho empieza a partir de la línea 140. Analicemos línea a línea.

Línea 140. Se asigna a la variable X el valor 21. Esta será la columna donde se desarrollará el movimiento.

Línea 150. Comienza un bucle en el cual se va variando el número de línea en la que se colocará la letra "O". Como la letra va a subir por la pantalla, el bucle va desde 20 hasta 1 con un incremento (STEP) de -1.

Líneas 160 y 170. Se imprime en la fila I + 1 un espacio en blanco. Como ocurrió en el programa anterior, en la primera vuelta del bucle esto no sirve para nada, pero a partir de la segunda nos borrará la posición anterior de la "O".

Líneas 180 y 190. Se imprime en la fila I la letra "O".

Líneas 200 y 210. Se hace un bucle en vacío para retardar el movimiento. Variando la longitud del bucle variará la velocidad.

Línea 210. En esta línea termina el bucle.

Cuando la letra "O" ha llegado a la fila número 1, ésta se borra gracias a las líneas 260 y 270.

Si has leído con atención lo que hemos dado hasta ahora te darás cuenta que los dos tipos de movimiento, horizontal y vertical, son muy parecidos. Si lo has entendido todo bien debes estar en condiciones de hacer tú mismo un programa que haga que un carácter se mueva por la pantalla de arriba hacia abajo. ¿Te atreves?

Espero que lo intentes. Por si no te apetece hacerlo te propongo que ejecutes el programa número 4 que te doy a continuación:

```
 10 REM ***************************************
 20 REM * MOVIMIENTO DE UNA 'O' A LO LARGO    *
 30 REM * DE UNA CARRETERA DE ARRIBA A ABAJO *
 40 REM ***************************************
 50 REM
 60 CLS
 70 FOR I=20 TO 1 STEP -1
 80    LOCATE I,19
 90    PRINT "!   !"
100 NEXT I
110 REM
120 REM *** MOVIMIENTO ***
130 REM
140 LET X=21
150 FOR I=2 TO 20
160    LOCATE I-1,X
170    PRINT " "
180    LOCATE I,X
190    PRINT "O"
200    FOR J=1 TO 100
210    NEXT J
220 NEXT I
230 REM
240 REM *** LA 'O' DESAPARECE ***
250 REM
260 LOCATE 20,X
270 PRINT " "
280 END
```

Si te fijas en el listado verás que lo único que he cambiado ha sido tres líneas. Intenta entender el significado de los cambios.

Respecto a las modificaciones para otros ordenadores son las mismas que hemos visto en los programas anteriores. También es conveniente que tú mismo aprendas a modificar los programas para que funcionen en tu ordenador. De todas maneras, nosotros te iremos dando todas las modificaciones necesarias para que puedas ejecutar los programas.

Movimiento diagonal en línea recta

Este tipo de movimiento no se diferencia mucho de los anteriores. En el movimiento horizontal variaba la columna. En el vertical la fila. En el movimiento diagonal varían, a la vez,

la fila y la columna. Esto hace que tengamos cuatro casos posibles de movimiento diagonal:

1. Positivo-positivo. La columna y la fila van aumentando de valor.

2. Positivo-negativo. La columna va aumentando mientras la fila va disminuyendo.

3. Negativo-positivo. La columna va disminuyendo mientras la fila va aumentando.

4. Negativo-negativo. La fila y la columna van disminuyendo.

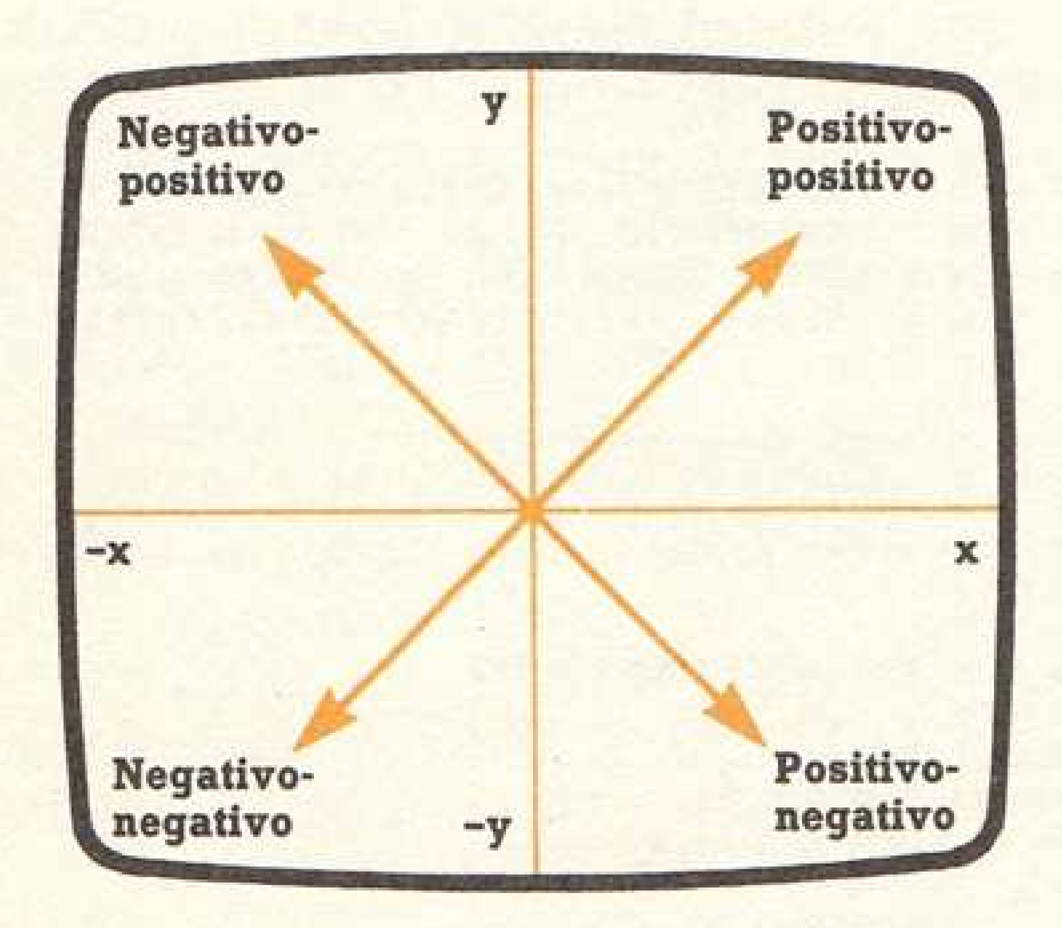


Fig. 4. Estos son los cuatro casos posibles de movimiento diagonal si los encuadramos en un eje de coordenadas.

Por otra parte, el movimiento diagonal puede ser más o menos pronunciado con respecto a la horizontal (o a la vertical). Esto es: el ángulo que forma la línea sobre la que se desarrolla el movimiento con la horizontal puede tener un valor comprendido entre 0 y 90 grados.

Un movimiento diagonal de 45 grados implicaría que a cada incremento de la columna le corresponde otro igual de la fila. Un ángulo de 90 grados implicaría que la fila iría variando, pero la columna no. Según esto, un ángulo de 0 grados implicaría que la columna iría variando, pero no la fila. Cualquier otro tipo de ángulo hace necesario hallar la relación entre el movimiento en la columna y en la fila.

Según lo dicho hasta ahora se deduce que los movimientos vertical y horizontal pueden ser tomados como dos casos particulares del movimiento diagonal. Si el ángulo es de 90 grados el movimiento será vertical. Si fuese de 0 grados éste sería horizontal.

Aunque todo esto nos pueda parecer un poco difícil de entender, no te preocupes. Con

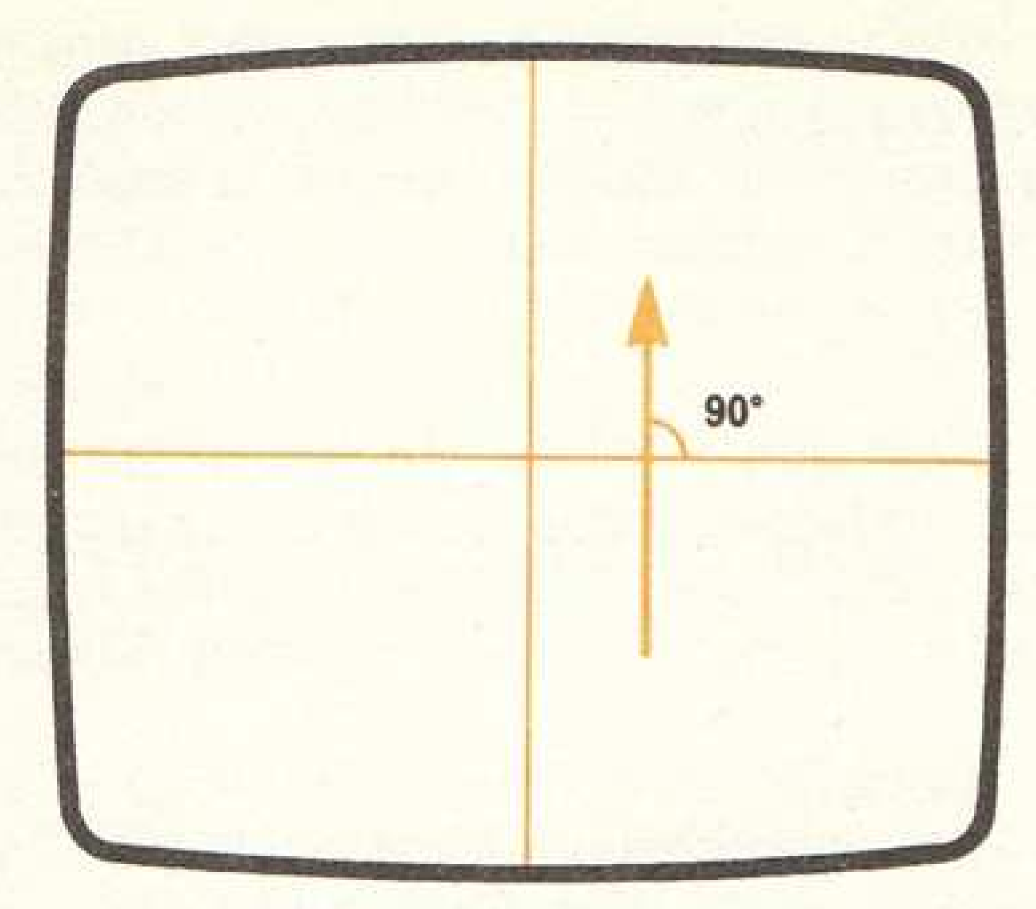


Fig. 5. El movimiento vertical es un movimiento diagonal con un ángulo de 90.°.

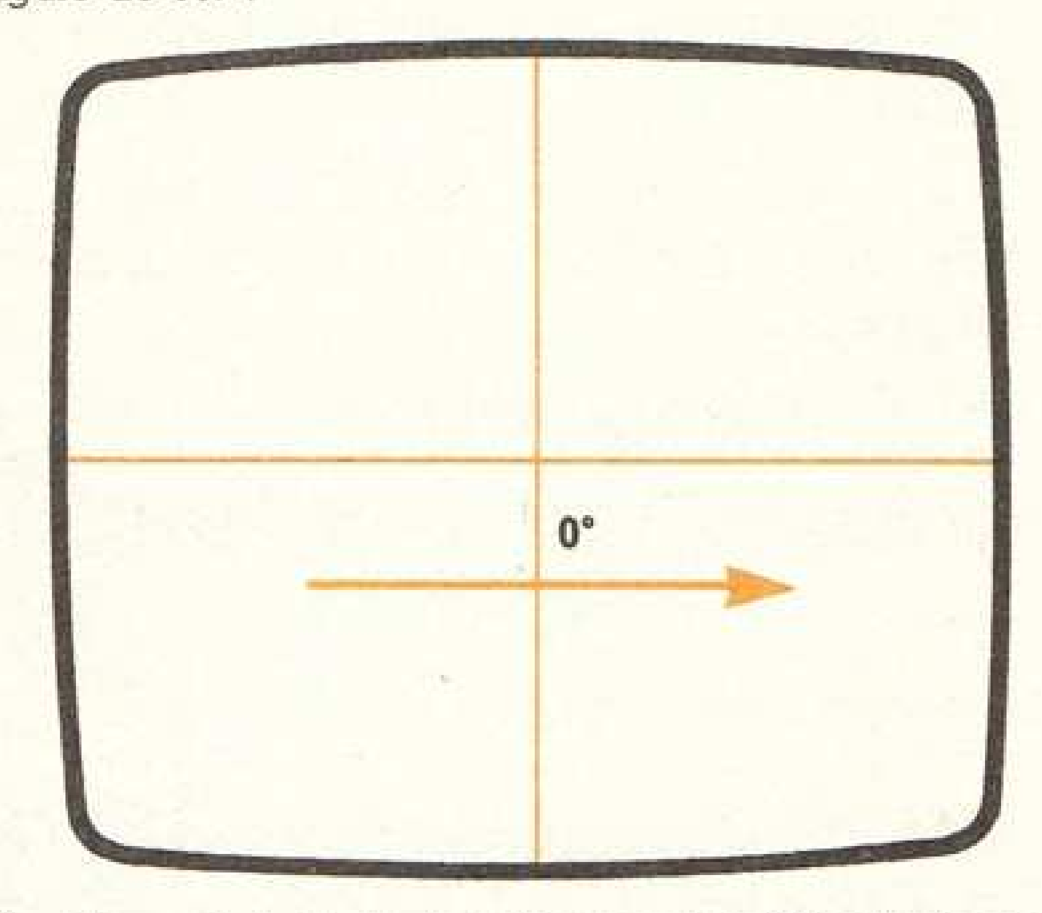
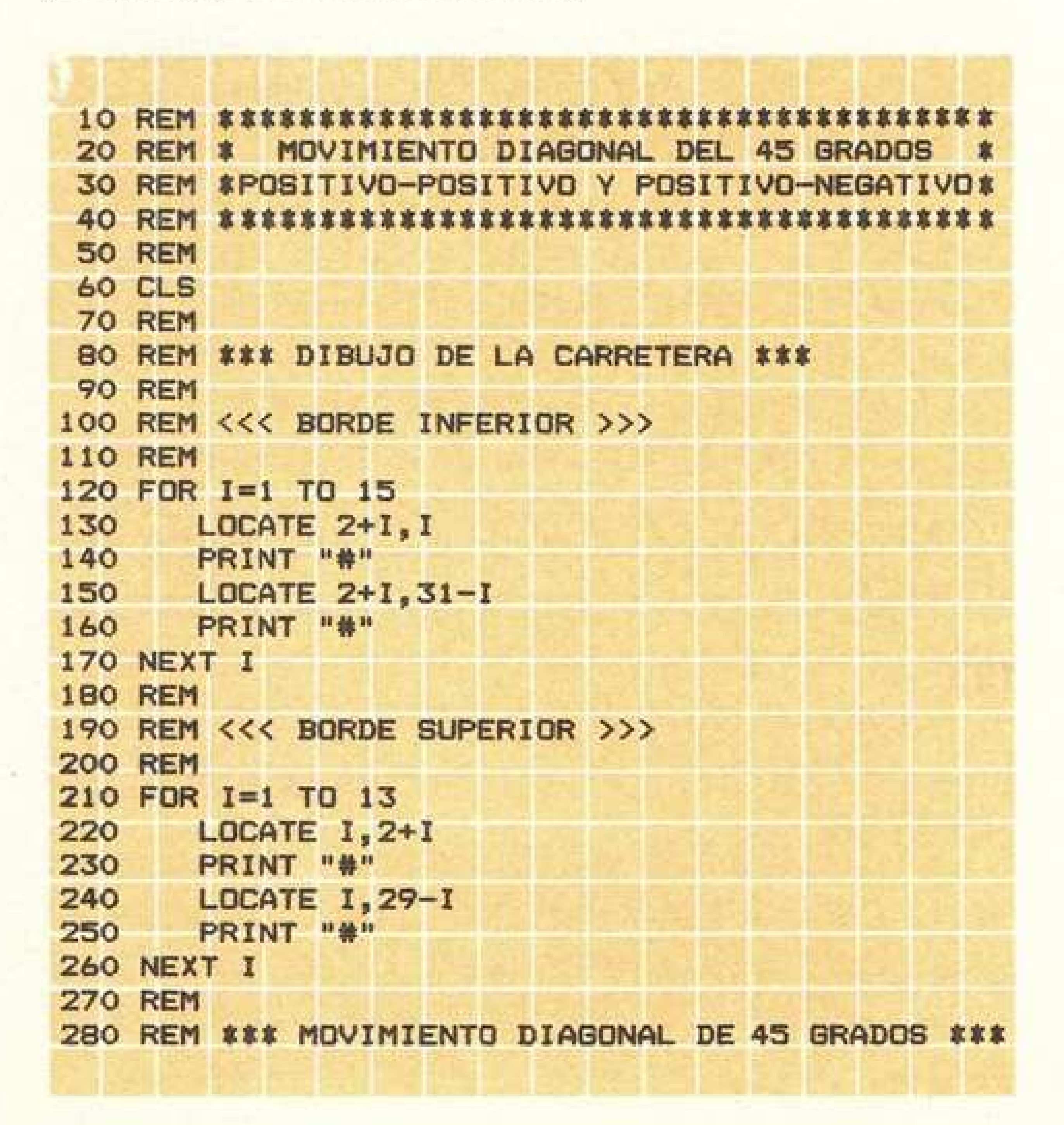
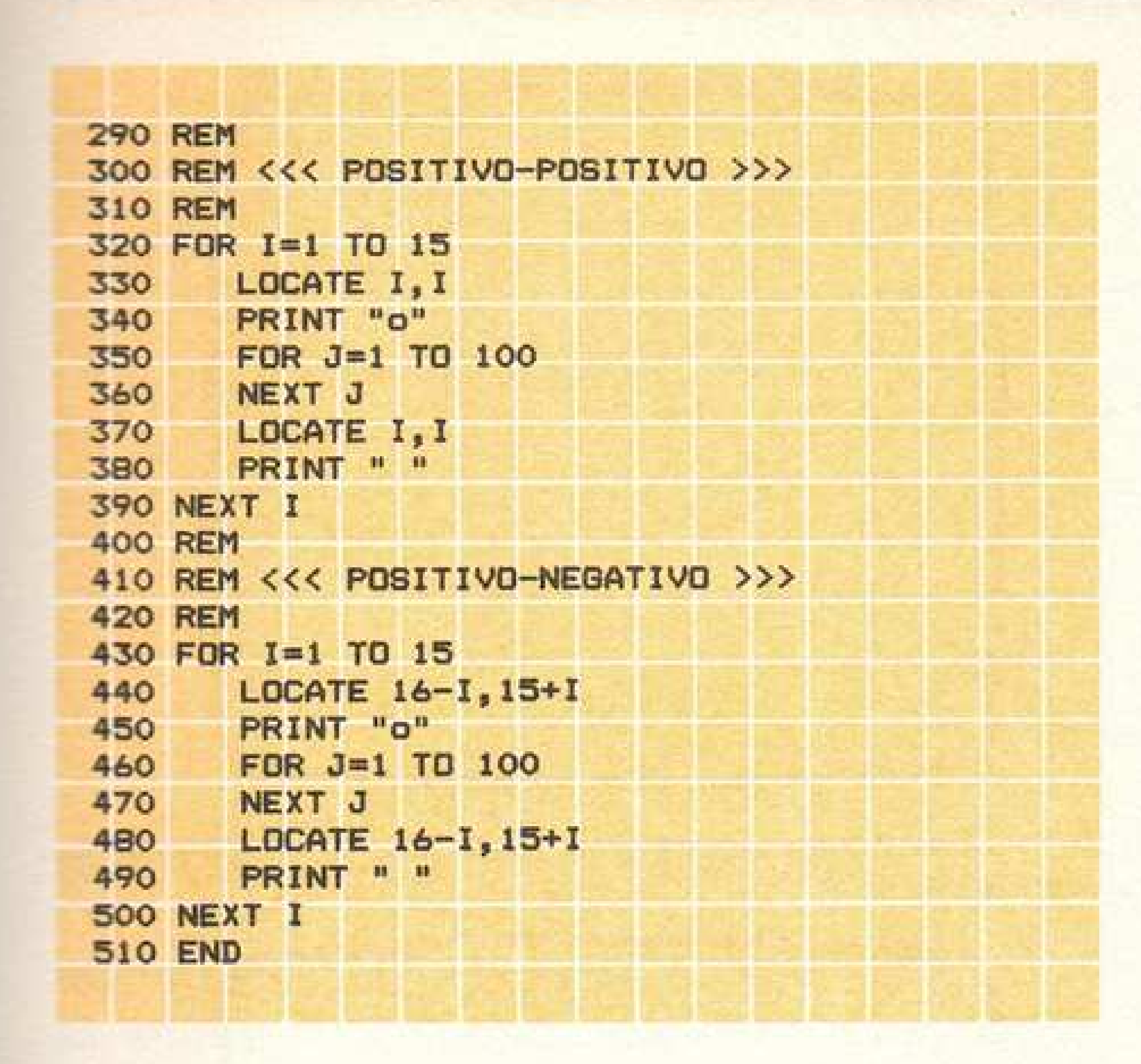


Fig. 6. El movimiento horizontal es un movimiento diagonal con un ángulo de 0°.

el programa que propongo a continuación verás que es más sencillo de lo que parece. El programa es el número 5.



```
10 REM ****************************************
20 REM *  MOVIMIENTO DIAGONAL DEL 45 GRADOS  *
30 REM *POSITIVO-POSITIVO Y POSITIVO-NEGATIVO*
40 REM ****************************************
50 REM
60 CLS
70 REM
80 REM *** DIBUJO DE LA CARRETERA ***
90 REM
100 REM <<< BORDE INFERIOR >>>
110 REM
120 FOR I=1 TO 15
130    LOCATE 2+I,I
140    PRINT "#"
150    LOCATE 2+I,31-I
160    PRINT "#"
170 NEXT I
180 REM
190 REM <<< BORDE SUPERIOR >>>
200 REM
210 FOR I=1 TO 13
220    LOCATE I,2+I
230    PRINT "#"
240    LOCATE I,29-I
250    PRINT "#"
260 NEXT I
270 REM
280 REM *** MOVIMIENTO DIAGONAL DE 45 GRADOS ***
```



```
290 REM
300 REM <<< POSITIVO-POSITIVO >>>
310 REM
320 FOR I=1 TO 15
330     LOCATE I,I
340     PRINT "o"
350     FOR J=1 TO 100
360     NEXT J
370     LOCATE I,I
380     PRINT " "
390 NEXT I
400 REM
410 REM <<< POSITIVO-NEGATIVO >>>
420 REM
430 FOR I=1 TO 15
440     LOCATE 16-I,15+I
450     PRINT "o"
460     FOR J=1 TO 100
470     NEXT J
480     LOCATE 16-I,15+I
490     PRINT " "
500 NEXT I
510 END
```

Este programa funciona sin problemas en el Amstrad e IBM. Se aconseja ejecutar este programa en el modo de 40 columnas por línea. Las modificaciones para los demás ordenadores son las siguientes:

COMMODORE

Línea 60 PRINT "<SHIFT-HOME>"

Todas las sentencias 'LOCATE' tienen que ser anuladas y sustituidas tal y como se ha visto en programas anteriores. La fórmula general es:

1. Asignar a la variable numérica Y el valor del primer argumento.
2. Asignar a la variable numérica X el valor del segundo argumento.
3. Hacer 'GOSUB 9900'

Si tomamos como ejemplo la línea 150:

150 LOCATE 2+I,31-I

nosotros tendríamos que poner:

150 Y=2+I:X=31-I:GOSUB 9900

Por supuesto, como se dijo más arriba, es necesario unir este programa a la rutina LOCATE PARA COMMODORE que se dio en el tomo 1.

MSX

Sólo hay que cambiar el orden de los argumentos de todas las sentencias LOCATE. Si la línea 150 es:

150 LOCATE 2+I,31-I

los usuarios del MSX tendrán que poner:

150 LOCATE 31-I,2+I

SPECTRUM

Hay que cambiar todas las sentencias LOCATE por 'PRINT AT'. Por ejemplo, la línea 150 nos quedaría:

150 PRINT AT 2+I,31-I;

No hay que olvidarse de poner el punto y coma (;) después de cada sentencia 'PRINT AT'.

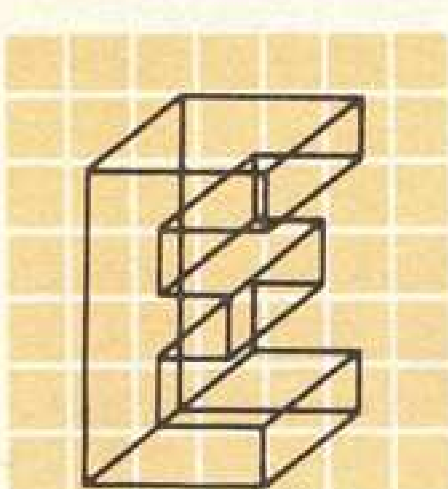
El funcionamiento del programa no reviste ninguna complicación. Lo único que se hace es variar a la vez la fila y la columna, incrementando ambas en uno, para realizar el movimiento. Te aconsejo pruebes a variar el programa para que, en vez de realizar un movimiento positivo-positivo y positivo-negativo, éste sea negativo-positivo y negativo-negativo.

Por si no te sale, el programa número 6 te muestra cómo hacerlo.

```
10 REM *******************************************
20 REM *    MOVIMIENTO DIAGONAL DEL 45 GRADOS    *
30 REM * NEGATIVO-POSITIVO Y NEGATIVO-NEGATIVO *
40 REM *******************************************
50 REM
60 CLS
70 REM
80 REM *** DIBUJO DE LA CARRETERA ***
90 REM
100 REM <<< BORDE INFERIOR >>>
110 REM
120 FOR I=1 TO 15
130     LOCATE 2+I,I
140     PRINT "#"
150     LOCATE 2+I,31-I
160     PRINT "#"
170 NEXT I
180 REM
190 REM <<< BORDE SUPERIOR >>>
200 REM
210 FOR I=1 TO 13
220     LOCATE I,2+I
230     PRINT "#"
240     LOCATE I,29-I
250     PRINT "#"
260 NEXT I
270 REM
280 REM *** MOVIMIENTO DIAGONAL DE 45 GRADOS ***
290 REM
300 REM <<< NEGATIVO-POSITIVO >>>
310 REM
320 FOR I=1 TO 15
330     LOCATE I,31-I
340     PRINT "o"
350     FOR J=1 TO 100
360     NEXT J
370     LOCATE I,31-I
380     PRINT " "
390 NEXT I
400 REM
410 REM <<< NEGATIVO-NEGATIVO >>>
420 REM
430 FOR I=1 TO 15
440     LOCATE 16-I,16-I
450     PRINT "o"
460     FOR J=1 TO 100
470     NEXT J
480     LOCATE 16-I,16-I
490     PRINT " "
500 NEXT I
510 END
```

Las modificaciones que hay que realizar son iguales a las del programa número 5.

Subrutina de petición de hora



N muchos programas es necesario, en un cierto momento, la introducción de la hora para que el programa que se está ejecutando en ese momento pueda seguir funcionando.

La rutina que proponemos a continuación nos pedirá la hora por pantalla. Dicha petición aparecerá en la fila y columna que deseemos, y la almacenará en la variable T$ con el siguiente formato:

hh:mm:ss

donde:

hh = horas
mm = minutos
ss = segundos

Para utilizar esta rutina necesitaremos la rutina de introducción datos que se dio en el tomo 1 y que comenzaba en la línea 9900.

Los parámetros que son necesarios pasarle a la rutina de entrada son los que se enumeran a continuación:

X = Columna donde se inicia la entrada de la hora.

Y = Fila donde se inicia la entrada de la hora.

SW = Variable que nos dice si se muestra la hora existente hasta el momento. Si SW = 0 no se muestra. Cualquier otro valor nos mostrará la hora actual.

El resultado de la rutina nos vendrá dado por la variable alfanumérica T$. En el caso que haya que mostrar la hora actual y T$ fuese una cadena vacía se mostrará la hora como:

00:00:00

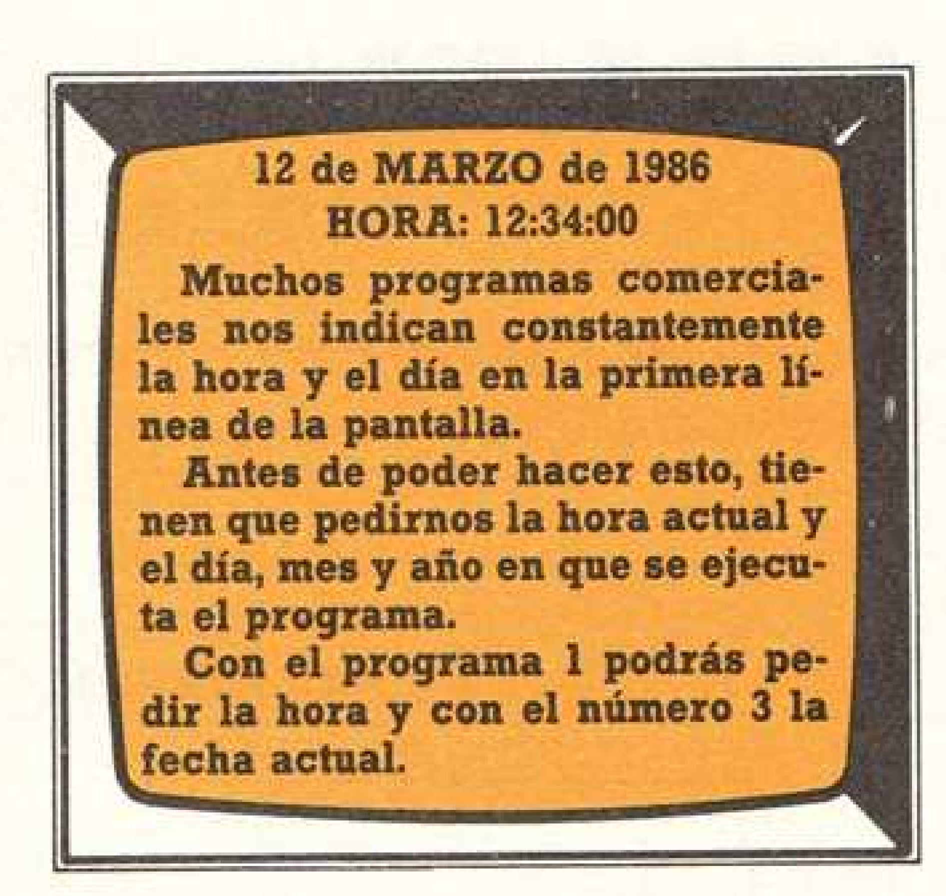
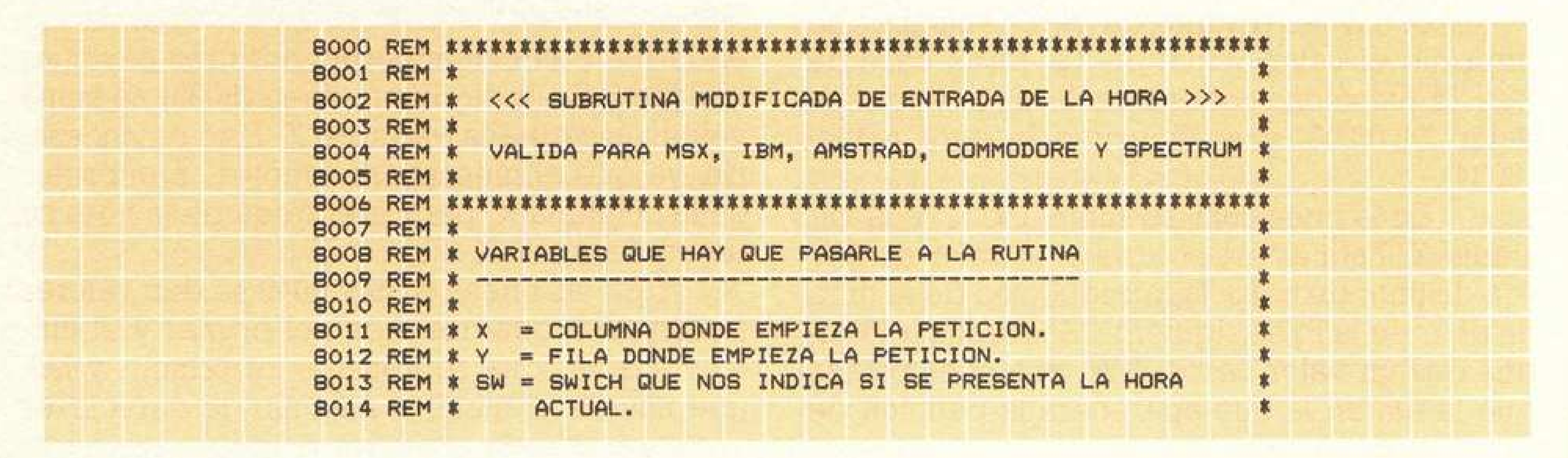


Fig. 1.

El funcionamiento del programa es el siguiente:



```
8000 REM ********************************************************
8001 REM *                                                      *
8002 REM *  <<< SUBRUTINA MODIFICADA DE ENTRADA DE LA HORA >>>  *
8003 REM *                                                      *
8004 REM *  VALIDA PARA MSX, IBM, AMSTRAD, COMMODORE Y SPECTRUM *
8005 REM *                                                      *
8006 REM ********************************************************
8007 REM *                                                      *
8008 REM * VARIABLES QUE HAY QUE PASARLE A LA RUTINA            *
8009 REM * -----------------------------------------            *
8010 REM *                                                      *
8011 REM * X  = COLUMNA DONDE EMPIEZA LA PETICION.              *
8012 REM * Y  = FILA DONDE EMPIEZA LA PETICION.                 *
8013 REM * SW = SWICH QUE NOS INDICA SI SE PRESENTA LA HORA     *
8014 REM *      ACTUAL.                                         *
```

```
8015 REM *                                                        *
8016 REM * EL RESULTADO SE RETORNA EN T$                          *
8017 REM *                                                        *
8018 REM * VARIABLES USADAS INTERNAMENTE POR LA RUTINA            *
8019 REM * -------------------------------------------            *
8020 REM *                                                        *
8021 REM * M$ = SIEMPRE TIENE VALOR "9"                           *
8022 REM * W$ = SIEMPRE TIENE VALOR "0"                           *
8023 REM * LO = SIEMPRE TIENE VALOR 2                             *
8024 REM * D$ = RESULTADO DE LA RUTINA DE ENTRADA DE DATOS        *
8025 REM *                                                        *
8026 REM *  ADEMAS DE ESTAS, SE UTLIZAN TODAS LAS USADAS POR LA   *
8027 REM * RUTINA DE ENTRADA DE DATOS.                            *
8028 REM *                                                        *
8029 REM **********************************************************
8030 REM
8031 IF SW=0 THEN GOTO 8035
8032 LOCATE Y,X
8033 IF T$="" THEN LET T$="00:00:00"
8034 PRINT "La hora actual es: ";T$
8035 LET Y=Y+1
8036 LOCATE Y,X
8037 PRINT "Introduzca la nueva hora  .. : .. : .."
8038 LET X=X+26
8039 M$="9":W$="0"
8040 LO=2
8041 GOSUB 9900
8042 IF VAL(D$)<0 OR VAL(D$)>24 THEN LET X=X-28:GOTO 8038
8043 IF D$="" THEN LET D$="00"
8044 LET T$=D$
8045 LET X=X+3
8046 GOSUB 9900
8047 IF VAL(D$)<0 OR VAL(D$)>59 THEN LET X=X-5:GOTO 8045
8048 IF D$="" THEN LET D$="00"
8049 LET T$=T$+":"+D$
8050 LET X=X+3
8051 GOSUB 9900
8052 IF VAL(D$)<0 OR VAL(D$)>59 THEN LET X=X-5:GOTO 8050
```

Modificaciones de los programas 1 y 2

Si cualquiera de los dos programas va a ser utilizado en una pantalla que tenga menos de 80 caracteres de ancho se propone variar las siguientes líneas:

```
8037 PRINT "Introduzca la nueva hora": LOCATE Y+1,X+5:PRINT "..:..:..":LET Y=Y+1
3088 X=X+5
3042 IF VAL (D$)<0 OR VAL (D$)>24 THEN LET X=X-7:GOTO 3038
```

En la línea 8031 se pregunta si la variable numérica SW es igual a cero (0). En caso afirmativo se transfiere el control del programa a la línea 8035. En esta línea empieza la recogida de la nueva hora.

Si SW es distinto de cero, entonces se imprime el valor actual de T$. En el caso de que T$ fuese una cadena vacía, se le asigna el valor '00:00:00' y se imprime la hora actual (8034).

En la línea 8035 se suma uno (1) a la variable Y. Esto es así porque si el valor de SW era distinto de cero, la introducción de la hora ha de ir en la línea siguiente. Si al utilizar la rutina das un valor cero a SW, entonces piensa que la fila en la que aparecerá la petición de datos no es la que tú le marcaste con la variable T, sino una línea más abajo.

Entre las líneas 8036 y 8037 se imprime el mensaje:

Introduzca nueva hora ..:..:..

en la fila y la columna que nosotros hayamos indicado. En la siguiente línea (8038) se suma veintiséis (26) a la variable X. Esto es necesario, ya que el mensaje se empieza a imprimir en la columna X, pero la entrada de la hora ha de empezar en X+26.

En las líneas 8039 y 8040 se dan valores a M$ y W$, que nos dicen el primer y el último carácter que se permite introducir, y LO, que nos da el número máximo de caracteres

que podemos introducir. Estos valores no se varían en todo el programa.

En la línea 9041 se llama a la subrutina de entrada de datos. Si, al retornar la rutina, el valor de D$ éste es mayor que 24 (veinticuatro horas hay en el día) o menor que 0 (no hay horas negativas), se vuelve a repetir la petición.

Si la entrada ha sido correcta, entonces se vuelca el contenido de D$ en T$ y se continúa con la petición de los minutos.

A partir de este momento el funcionamiento del programa es igual que lo que hemos visto anteriormente.

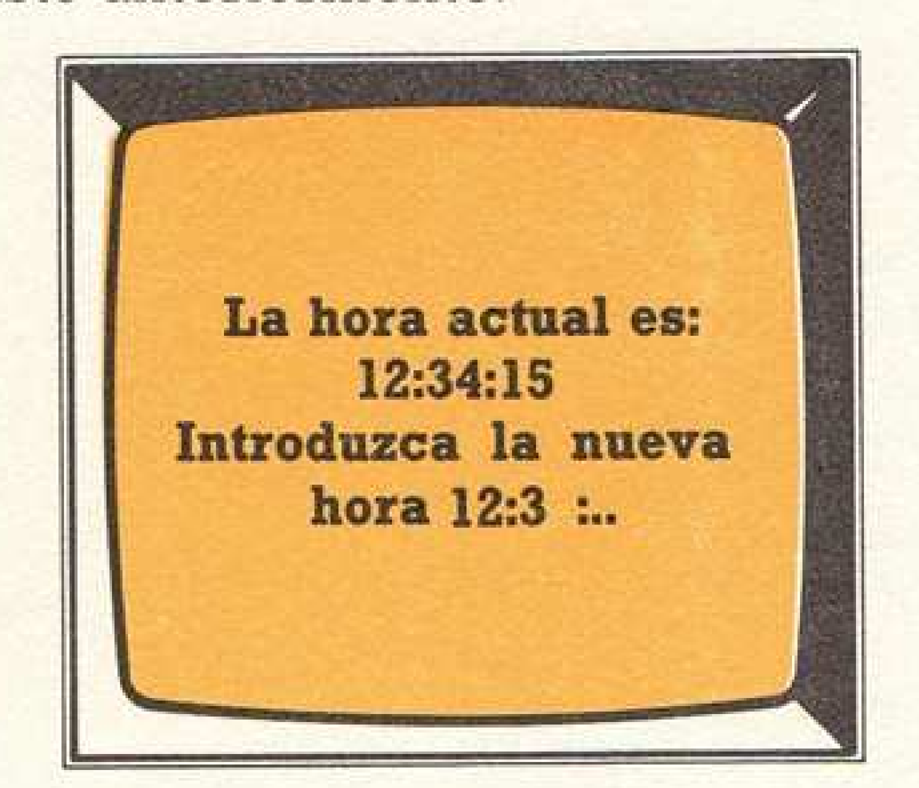


Fig. 2. Pantalla que aparece al ejecutar el programa 1.

Ejercicios resueltos

EJERCICIO N.º 1

En el programa 1, cuando se pulsa RETURN directamente, la variable D$ no coge ningún valor y VAL(D$) nos devuelve el valor cero (0). Debido a esto, si pulsamos en las tres entradas la tecla RETURN, no nos retornará, en D$, ningún valor, y T$ será igual a '::'.

¿Serías capaz de modificar el programa para que no ocurriese esto?

SOLUCION

Como puede apreciarse en el programa 2 lo único que hay que hacer es mirar si D$ es una cadena vacía (""). Si lo es, lo único que hay que hacer es que D$ sea igual a "00".

```
8000 REM ***********************************************************
8001 REM *                                                         *
8002 REM *          <<< SUBRUTINA DE ENTRADA DE LA HORA >>>        *
8003 REM *                                                         *
8004 REM *  VALIDA PARA MSX, IBM, AMSTRAD, COMMODORE Y SPECTRUM    *
8005 REM *                                                         *
8006 REM ***********************************************************
8007 REM *                                                         *
8008 REM * VARIABLES QUE HAY QUE PASARLE A LA RUTINA               *
8009 REM * ------------------------------------------              *
8010 REM *                                                         *
8011 REM * X  = COLUMNA DONDE EMPIEZA LA PETICION.                 *
8012 REM * Y  = FILA DONDE EMPIEZA LA PETICION.                    *
8013 REM * SW = SWICH QUE NOS INDICA SI SE PRESENTA LA HORA        *
8014 REM *      ACTUAL.                                            *
8015 REM *                                                         *
8016 REM * EL RESULTADO SE RETORNA EN T$                           *
8017 REM *                                                         *
8018 REM * VARIABLES USADAS INTERNAMENTE POR LA RUTINA             *
8019 REM * --------------------------------------------            *
8020 REM *                                                         *
8021 REM * M$ = SIEMPRE TIENE VALOR "9"                            *
8022 REM * W$ = SIEMPRE TIENE VALOR "0"                            *
8023 REM * LO = SIEMPRE TIENE VALOR 2                              *
8024 REM * D$ = RESULTADO DE LA RUTINA DE ENTRADA DE DATOS         *
8025 REM *                                                         *
8026 REM *  ADEMAS DE ESTAS, SE UTLIZAN TODAS LAS USADAS POR LA    *
8027 REM * RUTINA DE ENTRADA DE DATOS.                             *
8028 REM *                                                         *
8029 REM ***********************************************************
8030 REM
8031 IF SW=0 THEN GOTO 8035
8032 LOCATE Y,X
8033 IF T$="" THEN LET T$="00:00:00"
8034 PRINT "La hora actual es: ";T$
8035 LET Y=Y+1
8036 LOCATE Y,X
8037 PRINT "Introduzca la nueva hora  .. : .. : .."
8038 LET X=X+26
8039 M$="9":W$="0"
8040 LO=2
```

```
8041 GOSUB 9900
8042 IF VAL(D$)<0 OR VAL(D$)>24 THEN LET X=X-28:GOTO 8038
8043 LET T$=D$
8044 LET X=X+3
8045 GOSUB 9900
8046 IF VAL(D$)<0 OR VAL(D$)>59 THEN LET X=X-5:GOTO 8044
8047 LET T$=T$+":"+D$
8048 LET X=X+3
8049 GOSUB 9900
8050 IF VAL(D$)<0 OR VAL(D$)>59 THEN LET X=X-5:GOTO 8048
8051 LET T$=T$+":"+D$
8052 RETURN
8053 IF D$="" THEN LET D$="00"
8054 LET T$=T$+":"+D$
8055 RETURN
```

Subrutina de petición de fecha

Al igual que es necesario, según qué programas, tener una rutina para la petición de la hora, es necesario tener otra subrutina que le pida al usuario la fecha. Si estamos realizando un programa de contabilidad, necesitaremos saber en qué fecha se realizaron ciertos movimientos. Con esta rutina se puede realizar la entrada de cualquier fecha entre el año 1900 y el 1999. Aunque los años estén restringidos, se puede modificar el programa para que acepte fechas anteriores o posteriores a las prefijadas.

```
8100 REM ***********************************************************
8101 REM *                                                         *
8102 REM *        <<< SUBRUTINA DE ENTRADA DE LA FECHA >>>         *
8103 REM *                                                         *
8104 REM *  VALIDA PARA MSX, IBM, AMSTRAD, COMMODORE Y SPECTRUM    *
8105 REM *                                                         *
8106 REM ***********************************************************
8107 REM *                                                         *
8108 REM * VARIABLES QUE HAY QUE PASARLE A LA RUTINA               *
8109 REM * ----------------------------------------                *
8110 REM *                                                         *
8111 REM * X  = COLUMNA DONDE EMPIEZA LA PETICION.                 *
8112 REM * Y  = FILA DONDE EMPIEZA LA PETICION.                    *
8113 REM * SW = SWICH QUE NOS INDICA SI SE PRESENTA LA FECHA       *
8114 REM *      ACTUAL.                                            *
8115 REM *                                                         *
8116 REM * EL RESULTADO SE RETORNA EN F$                           *
8117 REM *                                                         *
8118 REM * VARIABLES USADAS INTERNAMENTE POR LA RUTINA             *
8119 REM * ------------------------------------------              *
8120 REM *                                                         *
8121 REM * M$ = SIEMPRE TIENE VALOR "9"                            *
8122 REM * W$ = SIEMPRE TIENE VALOR "0"                            *
8123 REM * LO = SIEMPRE TIENE VALOR 2                              *
8124 REM * D$ = RESULTADO DE LA RUTINA DE ENTRADA DE DATOS         *
8125 REM *                                                         *
8126 REM *  ADEMAS DE ESTAS, SE UTLIZAN TODAS LAS USADAS POR LA    *
8127 REM * RUTINA DE ENTRADA DE DATOS.                             *
8128 REM *                                                         *
8129 REM ***********************************************************
8130 REM
8131 IF SW=0 THEN GOTO 8135
8132 LOCATE Y,X
8133 IF F$="" THEN LET F$="01/01/1900"
8134 PRINT "Fecha actual: ";F$
8135 LET Y=Y+1
8136 LOCATE Y,X
8137 PRINT "Introduzca la nueza fecha  .. / .. / .."
8138 LET M$="9"
8139 LET W$="0"
8140 LET LO=2
8141 LET X=X+27
8142 GOSUB 9900
8143 IF VAL(D$)<0 OR VAL(D$)>31 THEN LET X=X-29:GOTO 8141
8144 LET F$=D$
8145 LET X=X+3
```

```
8146 GOSUB 9900
8147 IF VAL(D$)<1 OR VAL(D$)>12 THEN LET X=X-5:GOTO 8145
8148 LET F$=F$+"/"+D$
8149 LET X=X+3
8150 GOSUB 9900
8151 IF VAL(D$)<0 LET X=X-5:GOTO 8149
8152 IF D$="" THEN LET D$="00"
8153 LET F$=F$+"/19"+D$
8154 RETURN
```

Modificaciones al programa 3

Si el programa va a ser utilizado en una pantalla que tenga menos de 80 caracteres de ancho, se propone variar las siguientes líneas:

```
8137 PRINT "Introduzca fecha actual":LOCATE Y+1,X+5:PRINT "../../.."
8138 LET Y=Y+1
8141 LET X=X+5
8143 IF VAL (D$)<1 OR VAL(D$)>31 THEN LET X=X-7:GOTO 8141
```

El funcionamiento de este programa es muy parecido al anterior, pues incluso utiliza la misma rutina de entrada de datos que se dio en el tomo 1.

Fecha actual:
12/12/1996
Introduzca la nueva
fecha: 15/12/8_

Fig. 3. Pantalla que aparece al ejecutar el programa 3

Las variables que hay que pasarle para que funcione son:

X = columna donde empezará la petición de la fecha.

Y = fila donde empezará la petición de la fecha.

SW = Indica si se ha de imprimir la fecha que se tiene almacenada. Si SW es igual a cero, entonces no se muestra la fecha anterior. Si su valor es distinto de cero, se mostrará la fecha que se tiene almacenada. En el caso particular que no hubiese una fecha anterior almacenada, se mostraría en pantalla:

Fecha actual: 01/01/1900

El resultado de la entrada de la fecha aparece en la variable alfanumérica F$.

La forma de funcionar esta rutina es muy parecida a la anterior (Programa 1), con la única diferencia que el resultado de la variable T$ tiene el siguiente formato:

dd/mm/19aa

donde:

dd = día
mm = mes
19aa = año entre 1900 y 1999

Subrutina de conversión de la fecha

La siguiente subrutina transforma una cadena que contenga una fecha de la forma:

12/05/1987

en otra de cadena de la forma:

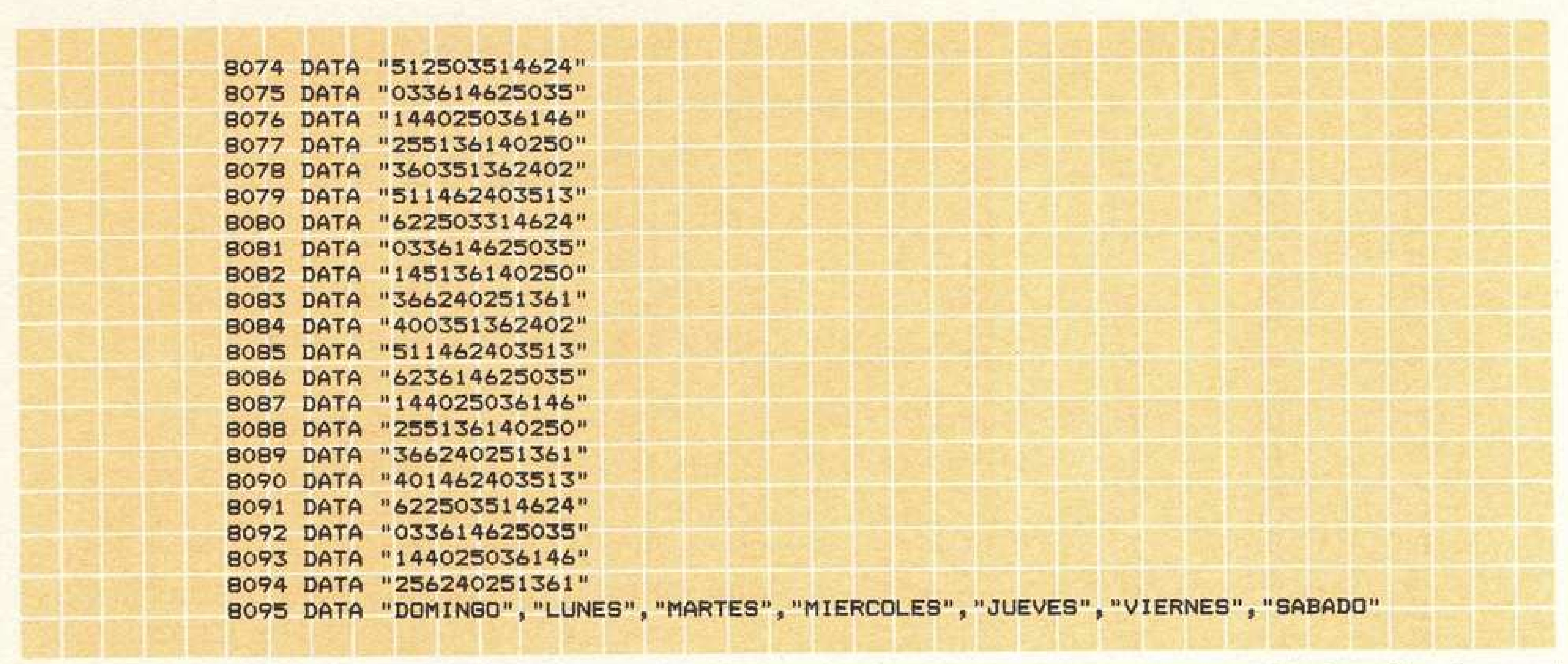
MARTES 12 de MAYO de 1987

Esta rutina es útil para hacer que nuestros programas sean más humanos y parezcan que entienden mejor nuestro lenguaje.



Fig. 4. Esta es una rutina que hace al ordenador un poco más cercano a nosotros.

```
8000 REM ************************************************************
8001 REM *                                                          *
8002 REM *         <<< PROGRAMA DE COMVERSION DE FECHA >>>          *
8003 REM *                                                          *
8004 REM *  VALIDO PAR MSX, IBM, AMSTRAD, COMMODORE Y SPECTRUM     *
8005 REM ************************************************************
8006 REM *                                                          *
8007 REM * VARIABLES QUE HAY QUE PASARLE A LA RUTINA                *
8008 REM * -----------------------------------------                *
8009 REM *                                                          *
8010 REM * F$ = FECHA A CONVERTIR. EL FORMATO A DE SER:             *
8011 REM *                                                          *
8012 REM *                         dd/mm/aaaa                       *
8013 REM *                                                          *
8014 REM * EL RESULTADO ES DEVUELTO EN R$                           *
8015 REM *                                                          *
8016 REM * VARIABLES USADAS INTERNAMENTE                            *
8017 REM * -----------------------------                            *
8018 REM *                                                          *
8019 REM * N$(1) = ALMACENA EL DIA                                  *
8020 REM * N$(2) = ALMACENA EL MES                                  *
8021 REM * N$(3) = ALMACENA EL AÑO                                  *
8022 REM * Z     = CONTADOR DE BUCLE                                *
8023 REM * A$    = VARIABLE LECTORA DE DATAS                        *
8024 REM * N     = VARIABLE AUXILIAR                                *
8025 REM * F     = VARIABLE CON LA FILA DE LA TABLA                 *
8026 REM * D     = CONTADOR DE LA TABLA DE DIAS DE LA SEMANA        *
8027 REM *                                                          *
8028 REM ************************************************************
8029 REM
8030 REM *** PROGRAMA PRINCIPAL ***
8031 REM
8032 IF F$="" THEN LET R$="SIN FECHA":RETURN
8033 DIM N$(3):REM <-- EN EL SPECTRUM CAMBIARLO POR: DIM N$(3,10)
8034 LET N$(1)=LEFT$(F$,2):REM <-- EN EL SPECTRUM CAMBIARLO POR: LET N$(1)=F$( T
O 2)
8035 LET N$(2)=MID$(F$,4,2):REM <-- EN EL SPECTRUM CAMBIARLO POR: LET N$(2)=F$(4
 TO 5)
8036 LET N$(3)=MID$(F$,7,4):REM <-- EN EL SPECTRUM CAMBIARLO POR: LET N$(3)=F$(7
 TO  10)
8037 LET R$=STR$(VAL(N$(1)))+" de "
8038 RESTORE 8066
8039 FOR Z=1 TO VAL(N$(2))
8040     READ A$
8041 NEXT Z
8042 LET R$=R$+A$+" de "+N$(3)
8043 REM
8044 REM *** CALCULO DEL DIA DE LA SEMANA ***
8045 REM
8046 LET N=VAL(MID$(F$,9,2))+4:REM <-- EN EL SPECTRUM CAMBIARLO POR: LET N=VAL(F
$(9 TO 10))+4
8047 LET F=(N/28-INT(N/28))*28+.5
8048 LET F=INT(F)
8049 RESTORE 8067
8050 FOR Z=1 TO F
8051     READ A$
8052 NEXT Z
8053 LET N=VAL(MID$(A$,VAL(N$(2)),1)):REM <-- EN EL SPECTRUM CAMBIARLO POR: LET
N=VAL(A$(VAL(N$(2))))
8054 LET D=N+VAL(N$(1))
8055 LET D=(D/7-INT(D/7))*7+.5
8056 IF D=0 THEN LET D=7
8057 RESTORE 8095
8058 FOR Z=1 TO D
8059     READ A$
8060 NEXT Z
8061 LET R$=A$+R$
8062 RETURN
8063 REM
8064 REM *** LINEAS DE DATAS ***
8065 REM
8066 DATA "ENERO","FEBRERO","MARZO","ABRIL","MAYO","JUNIO","JULIO","AGOSTO","SEP
TIEMBRE","OCTUBRE","NOVIEMBRE","DICIEMBRE"
8067 DATA "400351362402"
8068 DATA "511462403513"
8069 DATA "622503514624"
8070 DATA "034025036146"
8071 DATA "255136140250"
8072 DATA "366240251361"
8073 DATA "400351362402"
```



```
8074 DATA "512503514624"
8075 DATA "033614625035"
8076 DATA "144025036146"
8077 DATA "255136140250"
8078 DATA "360351362402"
8079 DATA "511462403513"
8080 DATA "622503314624"
8081 DATA "033614625035"
8082 DATA "145136140250"
8083 DATA "366240251361"
8084 DATA "400351362402"
8085 DATA "511462403513"
8086 DATA "623614625035"
8087 DATA "144025036146"
8088 DATA "255136140250"
8089 DATA "366240251361"
8090 DATA "401462403513"
8091 DATA "622503514624"
8092 DATA "033614625035"
8093 DATA "144025036146"
8094 DATA "256240251361"
8095 DATA "DOMINGO","LUNES","MARTES","MIERCOLES","JUEVES","VIERNES","SABADO"
```

Para utilizar esta rutina sólo hay que pasarle la fecha que se quiere procesar en la variable F$. La cadena resultante se devolverá en la variable alfanumérica R$.

Su funcionamiento es el siguiente:

Lo primero que se hace (línea 8032) es comprobar que la cadena F$ no es una cadena vacía. En el caso de que lo sea, se asignará a R$ el mensaje: "SIN FECHA".

En las líneas comprendidas entre la 8033 y 8036 se dimensiona la tabla alfanumérica N$ y se asigna a cada uno de sus elementos el día, el mes y el año, respectivamente. Esto se hace de esta manera para no tener que estar dividiendo el string F$ continuamente.

A continuación (línea 8037) se asigna a la variable R$ el número del día, quitando los ceros de la izquierda, seguido de la palabra: 'de'.

En las cuatro líneas siguientes (8038, 8039, 8040 y 8041) se lee de la línea de datas, que se encuentra en la línea 8066, el mes, en letra, de la fecha a modificar.

En la línea 8042 se concatena el valor actual de R$ con el mes leído y con el año de la fecha.

En este momento, si la variable F$ era igual a:

03/08/1944

nos encontramos con que R$ es igual a:

3 de AGOSTO de 1944

Ya sólo nos queda por conocer en qué día de la semana cayó el 3 de agosto de 1944.

A partir de este momento empieza la parte más complicada del programa. Antes de seguir con él, explicaré cómo se puede saber el día de la semana de cualquier día de cualquier año.

Más de una vez os habréis encontrado con unas tablas que, mediante una serie de sumas, nos permiten conocer en qué día de la semana cayó nuestro cumpleaños hace siete años. Estas tablas son como las que se pueden ver en la figura 5. Si, por ejemplo, queremos saber en qué día de la semana cayó el 12 de JUNIO de 1984, realizaremos los siguientes pasos:

1. Buscar en la tabla 'A' el año al que nos referimos.

2. Se busca en la tabla 'B' el número que se encuentra en la misma línea que el año y en la columna del mes. En el caso del ejemplo nos encontramos con el número 5.

3. Se suma el número que hemos hallado con el número de días. En nuestro caso: 12 + 5 = 17.

4. Se busca en la tabla 'C' el número resultante y se mira en qué fila se encuentra. En dicha fila está escrito el día de la semana. En nuestro ejemplo nos sale que el 12 de JUNIO de 1984 era MARTES.

TABLA 'A'			**1901-1999**
DOS ULTIMAS CIFRAS			
	25	**53**	**81**
	26	**54**	**82**
	27	**55**	**83**
	28	**56**	**84**
01	**29**	**57**	**85**
02	**30**	**58**	**86**
03	**31**	**59**	**87**

DOS ULTIMAS CIFRAS			
04	32	60	88
05	33	61	89
06	34	62	90
07	35	63	91
08	36	64	92
09	37	65	93
10	38	66	94
11	39	67	95
12	40	68	96
13	41	69	97
14	42	70	98
15	43	71	99
16	44	72	
17	45	73	
18	46	74	
19	47	75	
20	48	76	
21	49	77	
22	50	78	
23	51	79	
24	52	80	

TABLA 'B'

E	F	M	A	M	J	J	A	S	O	N	D
4	0	0	3	5	1	3	6	2	4	0	2
5	1	1	4	6	2	4	0	3	5	1	3
6	2	2	5	0	3	5	1	4	6	2	4
0	3	4	0	2	5	0	3	6	1	4	6
2	5	5	1	3	6	1	4	0	2	5	0
3	6	6	2	4	0	2	5	1	3	6	1
4	0	0	3	5	1	3	6	2	4	0	2
5	1	2	5	0	3	5	1	4	6	2	4
0	3	3	6	1	4	6	2	5	0	3	5
1	4	4	0	2	5	0	3	6	1	4	6
2	5	5	1	3	6	1	4	0	2	5	0
3	6	0	3	5	1	3	6	2	4	0	2
5	1	1	4	6	2	4	0	3	5	1	3
6	2	2	5	0	3	3	1	4	6	2	4
0	3	3	6	1	4	6	2	5	0	3	5
1	4	5	1	3	6	1	4	0	2	5	0
3	6	6	2	4	0	2	5	1	3	6	1
4	0	0	3	5	1	3	6	2	4	0	2
5	1	1	4	6	2	4	0	3	5	1	3
6	2	3	6	1	4	6	2	5	0	3	5
1	4	4	0	2	5	0	3	6	1	4	6
2	5	5	1	3	6	1	4	0	2	5	0
3	6	6	2	4	0	2	5	1	3	6	1
4	0	1	4	6	2	4	0	3	5	1	3
6	2	2	5	0	3	5	1	4	6	2	4
0	3	3	6	1	4	6	2	5	0	3	5
1	4	4	0	2	5	0	3	6	1	4	6
2	5	6	2	4	0	2	5	1	3	6	1

TABLA 'C'

D	1	8	15	22	29	36
L	2	9	16	23	30	37
M	3	10	17	24	31	
M	4	11	18	25	32	
J	5	12	19	26	33	
V	6	13	20	27	34	

Fig. 5. Tabla para hallar el día de la semana de una fecha.

Una vez conocido el método para encontrar el día de la semana es mucho más fácil continuar con la interpretación del programa.

Lo primero que se hace es asignar a la variable numérica N el valor de las dos últimas cifras del año más cuatro. Hay que sumarle cuatro porque el año 1901 está en la fila número 5 y esto impone un desplazamiento de: N.º de año + 4.

Una vez que tenemos inicializada la variable N asignamos a F la fila en la que se encuentra dicho año en la tabla 'A'. Esto se consigue quedándonos con el resto de dividir el año entre el número de filas de dicha tabla. Como en los ordenadores suele haber error a la hora de redondear, se le suma al resultado 0.5 para que no exista error.

A partir de la línea 8049 y hasta la 8052 se realiza un bucle, gracias al cual, cuando éste termina, almacenamos en A$ la fila que nos interesa.

Una vez que hemos encontrado la fila del año tenemos que encontrar la columna del mes. Pero la columna del mes coincide con el número de mes. Si el mes es ENERO, su columna es la número 1. Si el mes fuese JUNIO, su número sería el 6, y así sucesivamente.

Ahora sólo nos resta encontrar el número que se encuentra en la intersección de la columna del mes con la fila de dicho año. Sumamos dicho número con el número de día.

A continuación, y como pasó con la tabla 'A', dividimos el número entre siete y nos quedamos con el resto. Con esto llegamos a saber en qué día de la semana cayó la fecha que conocíamos.

Ejercicios resueltos

EJERCICIO N.º 1

En el programa 4 hemos visto cómo se puede representar la fecha de una manera más elegante a la habitual. Si te fijas en la figura 4 verás que el ordenador no sólo nos dice la fecha, sino que nos dice, además, si es por la mañana, por la tarde o por la noche. ¿Serías capaz de hacer un programa con el que se obtuviese el mismo resultado?

SOLUCION

```
1000 REM ***********************************************************
1010 REM *                                                         *
1020 REM *              <<< CONVERSION DE LA HORA 1 >>>            *
1030 REM *                                                         *
1040 REM * VALIDO PARA MSX, IBM, AMSTRAD, COMMODORE Y SPECTRUM     *
1050 REM ***********************************************************
1060 REM *                                                         *
1070 REM * VARIABLES QUE HAY QUE PASARLE A LA RUTINA               *
1080 REM * -----------------------------------------               *
1090 REM *                                                         *
1100 REM * T$ = HORA QUE SE QUIERE CONVERTIR. CON FORMATO:         *
1110 REM *                                                         *
1120 REM *                           hh:mm:ss                      *
1130 REM *                                                         *
1140 REM * EL RESULTADO SE RETORNA EN R$                           *
1150 REM *                                                         *
1160 REM * VARIABLES USADAS INTERNAMENTE                           *
1170 REM * -----------------------------                           *
1180 REM *                                                         *
1190 REM * A$ = VARIABLE AUXILIAR                                  *
1200 REM * N  = HORA A MODIFICAR                                   *
1210 REM * Z$ = HORA MODIFICADA                                    *
1220 REM *                                                         *
1230 REM ***********************************************************
1240 REM
8200 IF T$="" THEN LET R$="SIN HORA":RETURN
8201 IF MID$(T$,1,2)<"13" THEN LET A$="de la mañana":GOTO 8207:REM <-- EN EL SPE
CTRUM SUSTITUIR POR: IF T$( TO 2)<"13" THEN .........
8202 LET A$="de la tarde"
8203 LET N=VAL(MID$(T$,1,2))-12:REM <-- EN EL SPECTRUM SUSTITUIR POR: LET N=VAL(
T$( TO 2))-12
8204 IF N>8 THEN LET A$="de la noche"
8205 LET Z$=MID$(STR$(N),2,2):REM <-- EN EL SPECTRUM SUSTITUIR POR: LET Z$(STR$(
N))(2 TO 2)
8206 LET T$=Z$+MID$(T$,3):REM <-- EN EL SPECTRUM SUSTITUIR POR: LET T$=Z$+T$( 3
TO )
8207 LET R$=T$+" "+A$
8208 RETURN
```

Como puede verse la solución es muy sencilla. Consiste sólo en mirar si la hora es mayor de 12. Si lo es, entonces le restamos 12 a la hora y ponemos 'de la tarde'. Si fuese menor o igual, no restaríamos nada y pondríamos 'de la mañana'.

EJERCICIO N.º 2

Te imaginas que el ordenador te dijese:

Son las DOCE Y CUARTO pasadas

o bien:

Son las TRES menos 10

Hacer un programa que lo haga no es muy difícil. ¿Lo intentamos?

SOLUCION

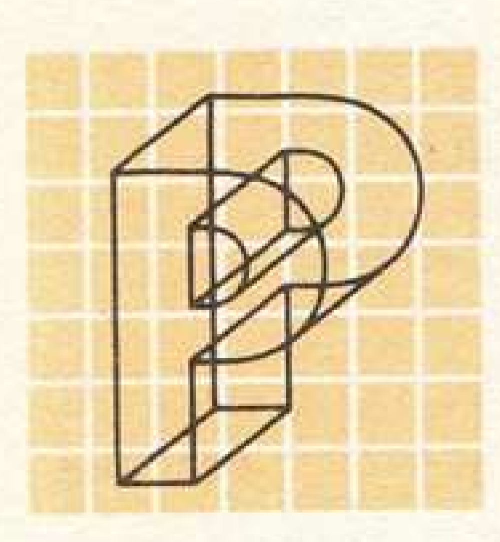
Esta vez no te digo cómo lo he hecho para que lo mires, te lo estudies y seas capaz de perfeccionarlo.

```
7600 REM ***********************************************************
7601 REM *                                                         *
7602 REM *              <<< CONVERSION DE LA HORA 2 >>>            *
7603 REM *                                                         *
7604 REM * VALIDO PARA MSX, IBM, AMSTRAD, COMMODORE Y SPECTRUM     *
7605 REM ***********************************************************
7606 REM *                                                         *
7607 REM * VARIABLES QUE HAY QUE PASARLE A LA RUTINA               *
7608 REM * -----------------------------------------               *
7609 REM *                                                         *
7610 REM * T$ = HORA QUE SE QUIERE CONVERTIR. CON FORMATO:         *
7611 REM *                                                         *
7612 REM *                           hh:mm:ss                      *
```

```
7613 REM *                                                      *
7614 REM * EL RESULTADO SE RETORNA EN R$                        *
7615 REM *                                                      *
7616 REM * VARIABLES USADAS INTERNAMENTE                        *
7617 REM * ------------------------------                       *
7618 REM *                                                      *
7619 REM * H$ = VARIABLE QUE ALMACENA LA HORA                   *
7620 REM * M$ = VARIABLE QUE ALMACENA LOS MINUTOS               *
7621 REM * H  = VARIABLE QUE ALMACENA LA HORA                   *
7622 REM * M  = VARIABLE QUE ALMACENA LOS MINUTOS               *
7623 REM * A$ = NOS DICE SI ES POR LA MAÑANA, TARDE O NOCHE     *
7624 REM * B$ = NOS DICE SI ES POSITIVA O NEGATIVA LA HORA      *
7625 REM * C$ = "Son la(s) "                                    *
7626 REM * D$ = NOS DICE SI LAS HORAS SON PASADAS               *
7627 REM * Z  = CONTADOR DE BUCLE                               *
7628 REM *                                                      *
7629 REM ********************************************************
7630 REM
7631 IF T$="" THEN LET R$="SIN HORA":RETURN
7632 LET A$=" de la mañana"
7633 LET H$=LEFT$(T$,2):LET H=VAL(H$)
7634 LET M$=MID$(T$,4,2):LET M=VAL(M$)
7635 IF H>12 THEN LET H=H-12:LET A$=" de la tarde":IF H>8 THEN LET A$=" de la no
che"
7636 LET B$=" y ":LET C$="Son la(s) "
7637 IF M>34 THEN H=H+1+12*(H=12):M=60-M:LET B$=" menos "
7638 RESTORE 7642
7639 FOR Z=1 TO H
7640    READ H$
7641 NEXT Z
7642 DATA "UNA","DOS","TRES","CUATRO","CINCO","SEIS","SIETE","OCHO","NUEVE","DIE
Z","ONCE","DOCE"
7643 LET R$=C$+H$+B$
7644 LET D$=" pasadas"
7645 IF M/5=INT(M/5) OR LEFT$(R$,1)<>"S" THEN D$=""
7646 LET M=INT(M/5)*5-5*(B$=" menos " AND (M/5<>INT(M/5))):LET M$=STR$(M)
7647 IF M$=" 30" THEN LET M$="media"
7648 IF M$=" 15" THEN LET M$="cuarto"
7649 LET R$=R$+M$+D$+A$
7650 RETURN
```

EL TALLER DEL HARDWARE

Fases del desarrollo del primer proyecto



PARA los ya iniciados en la construcción de prototipos, con las indicaciones apuntadas en los dos primeros apartados es posible que fueran capaces de realizar el montaje de la tarjeta de ampliación de puertos de entrada/salida sin ningún contratiempo. Para los que comienzan a lidiar en esta arena vamos a describir con más detalle cómo se diseñan y se realizan los montajes de prueba en el típico taller que todo buen aficionado a la informática y a la electrónica debería tener. En un número próximo describiremos también cómo montarse el taller y de qué herramientas disponer.

El diseño comienza con la descripción orgánica y funcional del dispositivo que deseamos montar, empleando para ello diagramas de bloques y dibujos de las conexiones lógicas entre los componentes. Después nos planteamos sobre qué sistema físico lo vamos a montar: circuito sin soldaduras, circuitos de tiras, circuito impreso o cualquier tipo de montaje con el que tengamos práctica y del que dispongamos de los medios materiales. Conviene también hacer un plano físico de la colocación de los componentes, cortes, hilos, conectores y demás detalles. Después se realizan las conexiones oportunas, la soldadura de los circuitos y la fijación de las partes mecánicas. A continuación es necesario verificar que lo montado corresponde a lo que se describió en los dibujos teóricos, comprobando las conexiones una por una para luego no llevarse sorpresas. Se verifican especialmente las conexiones a las alimentaciones y a las líneas de conexión a los buses, pues estas suelen ser las partes más delicadas para el funcionamiento general del sistema. Comprobadas concienzudamente las conexiones llega la hora de la verdad y enchufamos el equipo al ordenador, estando todo apagado, por supuesto. Aplicamos tensión y en principio todo tiene que funcionar como siempre. Cargamos un programa de tests de la parte añadida y verificamos dinámicamente cada una de las señales de la interfaz, hasta conseguir los niveles o las formas de onda apropiadas.

Veamos en el ejemplo propuesto de la tarjeta de ampliación de puerto de entrada/salida cómo se realiza cada una de las fases. Daremos los detalles y las fotografías de la realización para ZX-Spectrum, pero en su mayor parte es válido para los otros ordenadores. Solamente es necesario cambiar la parte de decodificación de direcciones.

Para la tarjeta de ampliación de puertos el diagrama es muy simple:

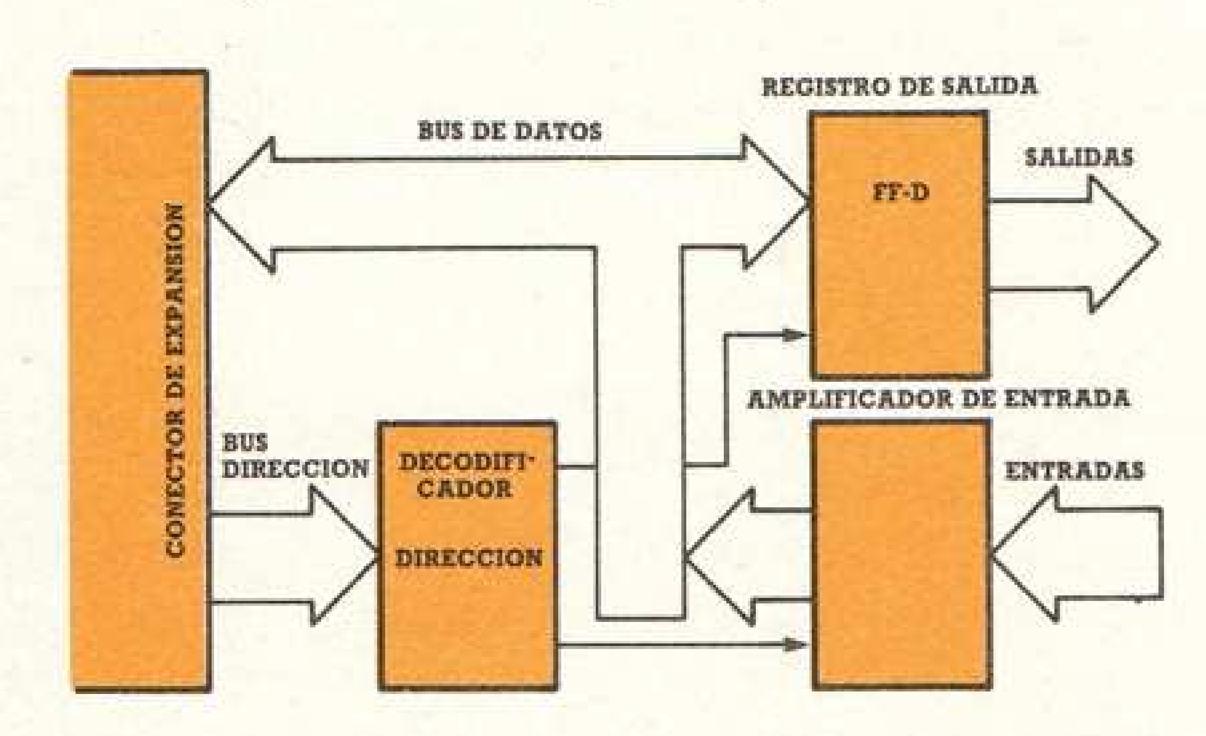


Fig. 1. Diagrama de bloques de la tarjeta ampliación de puertos E/S.

En el diagrama de bloques podemos distinguir:

El conector de expansión del ordenador. Es diferente para cada una de las máquinas. Presenta todas las señales necesarias para poder ampliar el número de dispositivos.

Bus de datos. Es un conjunto de 8 hilos dentro del conector de expansión que propaga los datos entre las diferentes unidades. Es de tipo bidireccional.

Bus de dirección. Es un conjunto de 16 a 20 hilos dentro del conector de expansión que presenta la dirección del dispositivo al que se dirige la unidad central en sus transferencias.

Decodificador de dirección. Determina cuándo un dispositivo ha sido direccionado por la unidad central, produciendo un pulso, que se emplea para activar la transferencia.

Registro de salida. Formado mediante biestables de tipo D, se cargan con la información del bus de datos al producirse en el bus de direcciones la dirección del puerto de salida.

Amplificadores de entrada. Formado mediante puertas de tres estados, transfieren al bus de datos la información existente en sus entradas al producirse en el bus de direcciones la dirección del puerto de entrada.

Sobre este dibujo se suele también marcar el número de hilos de cada bus de señales, el sentido de las transferencias y a veces alguna característica particular del diseño, como puede ser un pulso de sincronismo exterior o una señal de referencia.

Se suele denominar diagrama lógico para diferenciarlo del esquema real, que suele denominarse diagrama físico. En este diagrama lógico describimos las conexiones en su forma teórica, cómo deberían ser. Describimos cada componente mediante un símbolo que pueda ser interpretado mediante un golpe de vista por cualquiera que conozca este "lenguaje". Hay varias escuelas y normas de descripción de componentes que pretenden ser los más descriptivos. Para los componentes sencillos como puertas y biestables hay casi acuerdo, siendo los símbolos más usados los propuestos por el IEEE y son los que usaremos en nuestras descripciones. Para los componentes de alta integración usaremos casi siempre bloques rectangulares con las entradas a la izquierda, las salidas a la derecha y las señales de control y alimentación en la parte superior o inferior según resulte más apropiado. Es importante recalcar que los diagramas deben ser usados para describir y ayudar a entender los circuitos, por lo que todo lo que permita una descripción más clara deberá recalcarse y todo lo que pueda confundir deberá suprimirse o presentarse en otro dibujo adicional. Por desgracia hay más excepciones que reglas, por lo que muchas veces resulta arduo el entender lo que un circuito significa, aunque vaya ilustrado con profusión de comentarios.

En general, las líneas que se cruzan no llevan conexión eléctrica a menos que se indique mediante un punto negro.

El diagrama lógico ya fue presentado en el primer capítulo, por lo que lo reproducimos aquí a menor tamaño.

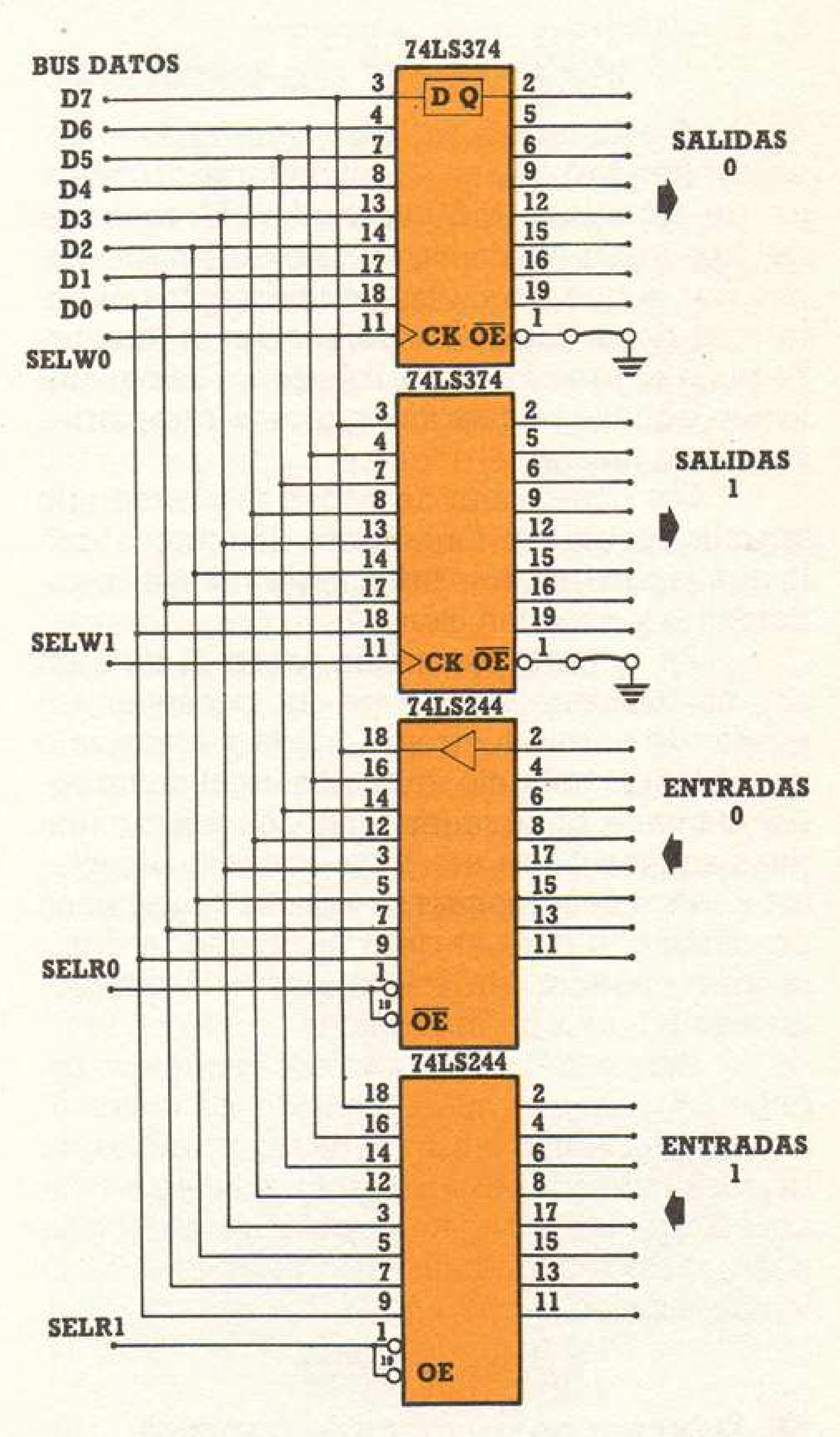


Fig. 2. Diagrama lógico de la tarjeta de ampliación de puertos de E/S.

Para un montaje rápido y que permita realizar cambios fácilmente se emplean tarjetas como la de la fotografía.

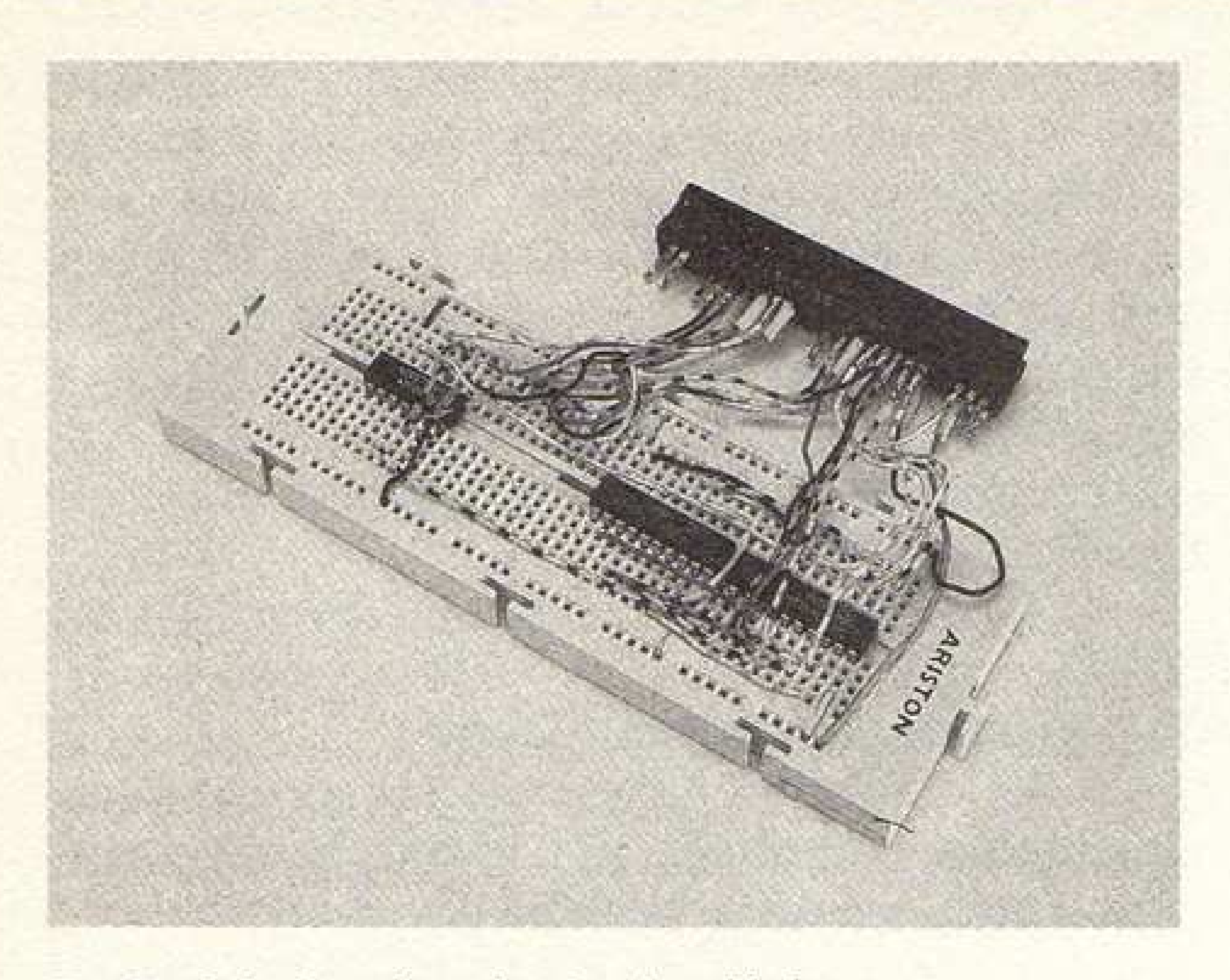


Fig. 3. Solución sobre circuito sin soldaduras.

A esta tarjeta se le ha conectado el conector hembra que ha de enchufarse al conector de expansión del ordenador. El conector del Spectrum presenta señales en las dos caras, por lo que es necesario utilizar un conector con doble fila de contactos. En la fotografía puede apreciarse que solamente se han utilizado aquellos contactos que son necesarios para esta montaje.

Las conexiones se hacen mediante hilo sencillo recubierto de plástico de colores, con lo que la identificación se puede realizar rápidamente y sin confusión.

En la tarjeta montada sobre el circuito sin soldaduras, solamente se presenta un puerto de entrada, otro de salida y el circuito de decodificación de dirección. Realmente sería de poca utilidad montar todo el circuito pues solamente es util para verificar el sistema cuando sea la primera vez que ensayamos un circuito, o para ver sus posibilidades límite, como pueden ser los valores de consumo, niveles o forma de las señales.

Para este tipo de montajes podemos dibujar el plano con indicación de las conexiones de entrada y salida para recordatorio de los resultados del experimento, pues la tarjeta necesitaremos utilizarla en otros ensayos, una vez que hayamos trasladado el circuito a un montaje definitivo.

Solución sobre circuito impreso de tiras

Empezaremos por dibujar el esquema sobre papel cuadriculado, a ser posible a escala doble aproximadamente, para trabajar más cómodamente, pero sin olvidar dibujar todos los elementos a escala.

Marcamos las posiciones de los circuitos integrados y de los demás componentes mediante puntos negros: mediante un símbolo especial indicaremos la patilla número 1. Las tiras que deben ser cortadas, por pertenecer a nodos diferentes, las marcaremos con un aspa.

Las líneas de conexión horizontal se suponen que son las tiras propias del circuito, por lo que no se marcarán. Anotaremos los

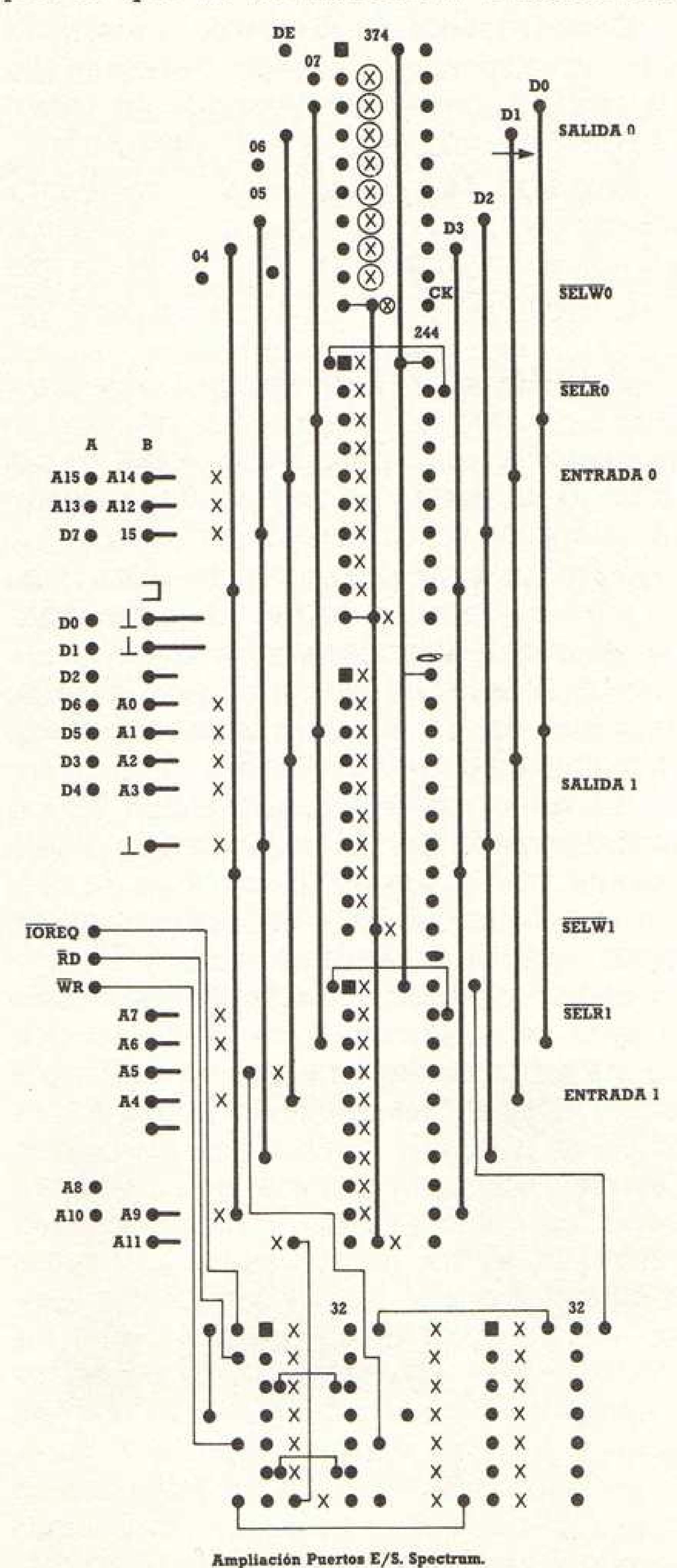


Fig. 4. Dibujo de las conexiones para circuito de tiras.

nombres de las señales principales para facilitar la revisión y el seguimiento de las pruebas.

El dibujo se suele hacer normalmente considerando que se ve desde la parte de los componentes y, por tanto, del lado de los puentes. Hay que tenerlo presente a la hora de realizar los cortes entre pistas, pues al cambiar de lado podríamos cortar donde no corresponde. Para facilitar la identificación del punto, pueden marcarse con rotulador los puntos donde corresponda cortar y luego cortar todos a la vez.

Los cortes se realizan perfectamente con una broca de 4 a 5 milímetros de diámetro a la que se añade un mango para poder manejarla mejor.

Las conexiones entre pistas paralelas se realizarán mediante puentes. Si la línea de conexión es vertical, el hilo puede ser desnudo. Si la línea de conexión es entre puntos que no están en la misma vertical, deberá usarse hilo aislado. La conexión puede hacerse por el lado de los componentes o por el de las soldaduras, pero es mejor por el lado de los componentes para dejar diáfano el otro lado y facilitar las soldaduras. Las conexiones utilizando hilo vertical presentan una apariencia mejor, pero requieren mucho más espacio que si usamos hilo aislado. Además, los electrones saben buscar su camino, aunque no sea recto.

En la fotografía puede verse la solución sobre circuito de tiras de la tarjeta de ampliación de puertos de E/S.

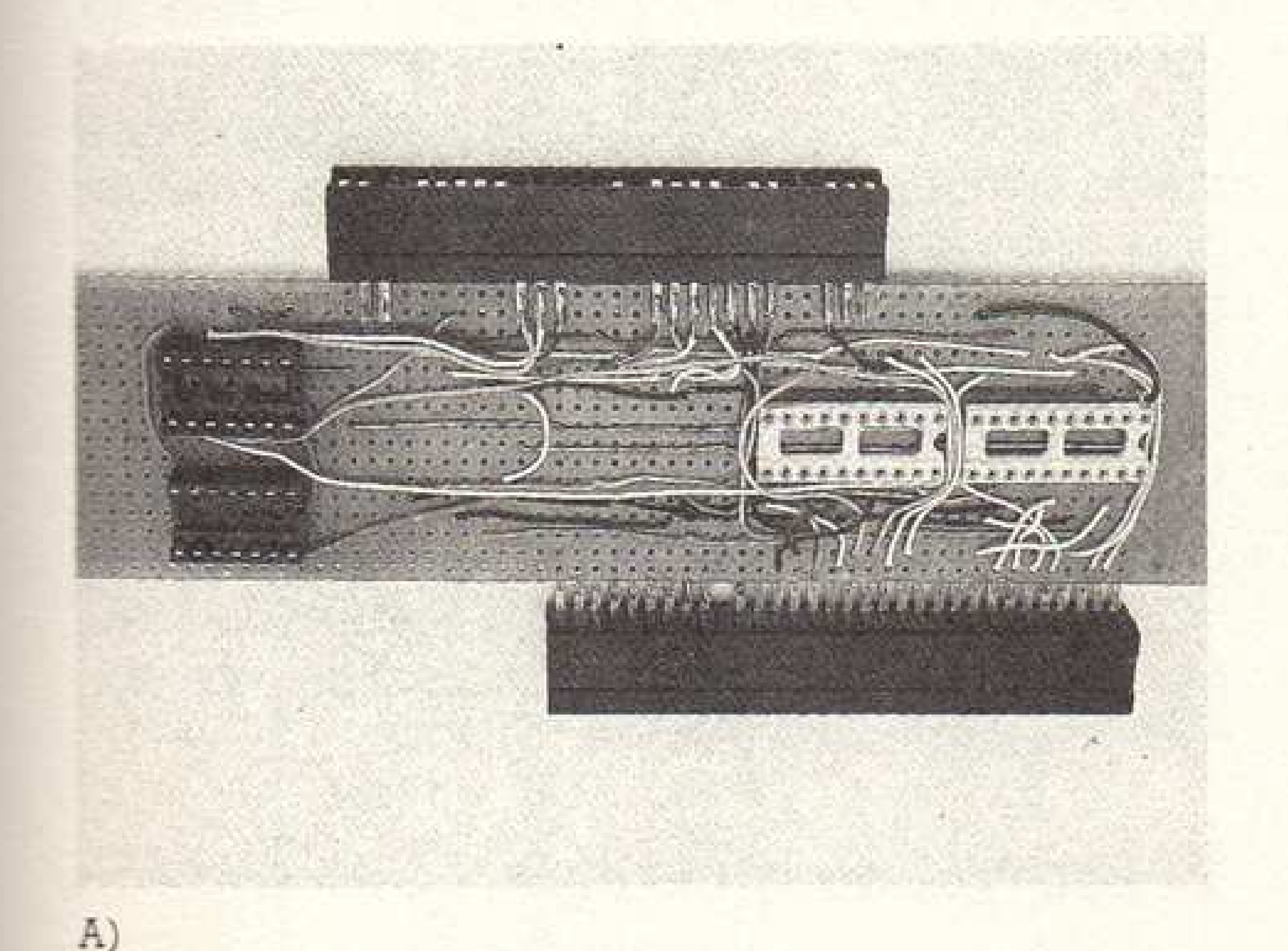
Puede apreciarse cómo va soldado el conector mediante los terminales inferiores. Las señales de la fila superior de contactos se toman a través de hilos directamente desde el conector. Solamente se han colocado los zócalos de los puertos de arriba para permitir observar la colocación de los hilos de alimentación y masa por en medio de las patillas de los circuitos integrados.

Este tipo de montaje, como puede apreciarse, es muy laborioso y solamente se justifica para prototipos únicos. La placa de circuito impreso de tiras se ha cortado de un placa de mayor longitud para contener los circuitos necesarios para este montaje. Una sierra de marquetería es idónea para cortar el circuito impreso. Hay otras presentaciones de circuitos previamente cortados, dejando grupos de tres agujeros contiguos unidos.

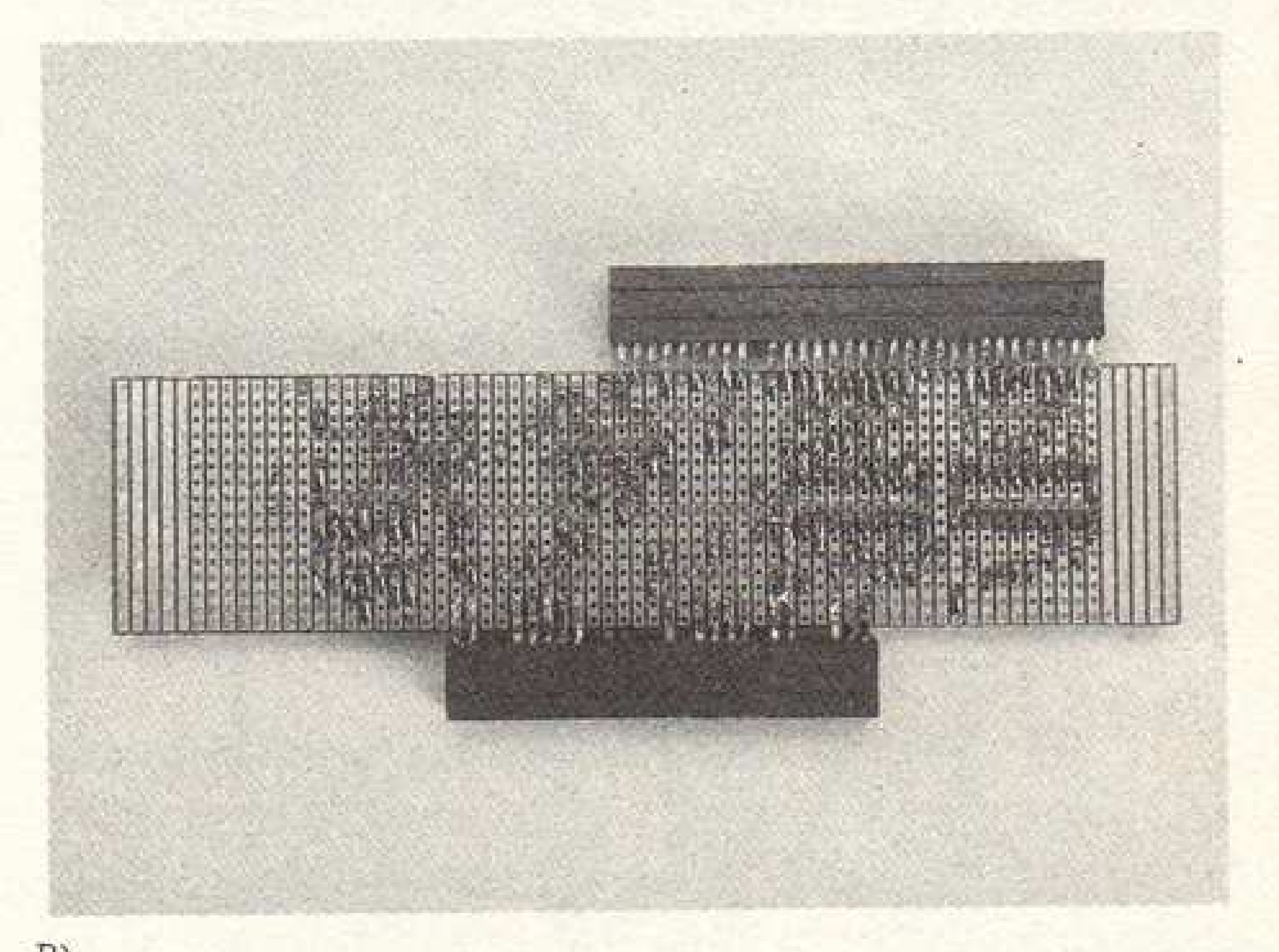
Solución con hilo enrollado

Para montajes fiables y de gran duración, cuando el número de unidades no es muy elevado se emplean tarjetas de circuito impreso con perforaciones en todos los puntos donde puedan ubicarse patillas de circuito. Se introducen zócalos de circuito integrado de patillas muy larga, de sección cuadrangular con los cantos vivos. Sobre estas patillas se enrollan los hilos con los que se hacen las conexiones.

Para el desarrollo de circuitos con esta técnica se comienza por describir las conexiones mediante un diagrama lógico, como ya vimos antes. Luego se distribuyen los circuitos integrados sobre la tarjeta de forma que las distancias entre componentes resulten mínimas. Para esta fase es corriente emplear el ordenador, pues fácilmente puede realizar lo que a nosotros nos llevaría horas: calcular la longitud del cable para realizar todas las co-



A)



B)

Fig. 5. Solución sobre circuito impreso de tiras. Se muestran ambas caras.

nexiones para una ubicación de componentes dada. Por esto es frecuente realizar una tabla de conexiones indicando:

— *Nombre de la señal.*
— *Pastilla y pata origen.*
— *Pastilla y pata destino.*
— *Número de secuencia dentro de la señal.*
— *Longitud de la conexión.*

Con esta tabla introducida en el ordenador, la tabla de ubicación de componentes y la descripción geométrica de cada componente, el ordenador puede ayudarnos a mejorar la distribución de componentes de un diseño.

Daremos más detalles cuando tratemos el tema de diseño de circuitos impresos por ordenador. Hay máquinas que a partir de las tablas indicadas realizan el montaje con hilos, utilizando para ello una técnica similar a la empleada en los registradores gráficos: la máquina toma el hilo de la longitud indicada, coloca el cabezal frente al punto origen, enrolla el primer terminal, desplaza el cabezal hasta el punto destino y enrolla el segundo terminal. El cabezal eléctrico es bastante costoso. Puede realizarse el enrollamiento mediante una herramienta manual, pero al no poder dar una fuerza de apriete uniforme, el resultado no es igual que mediante la máquina de enrollar.

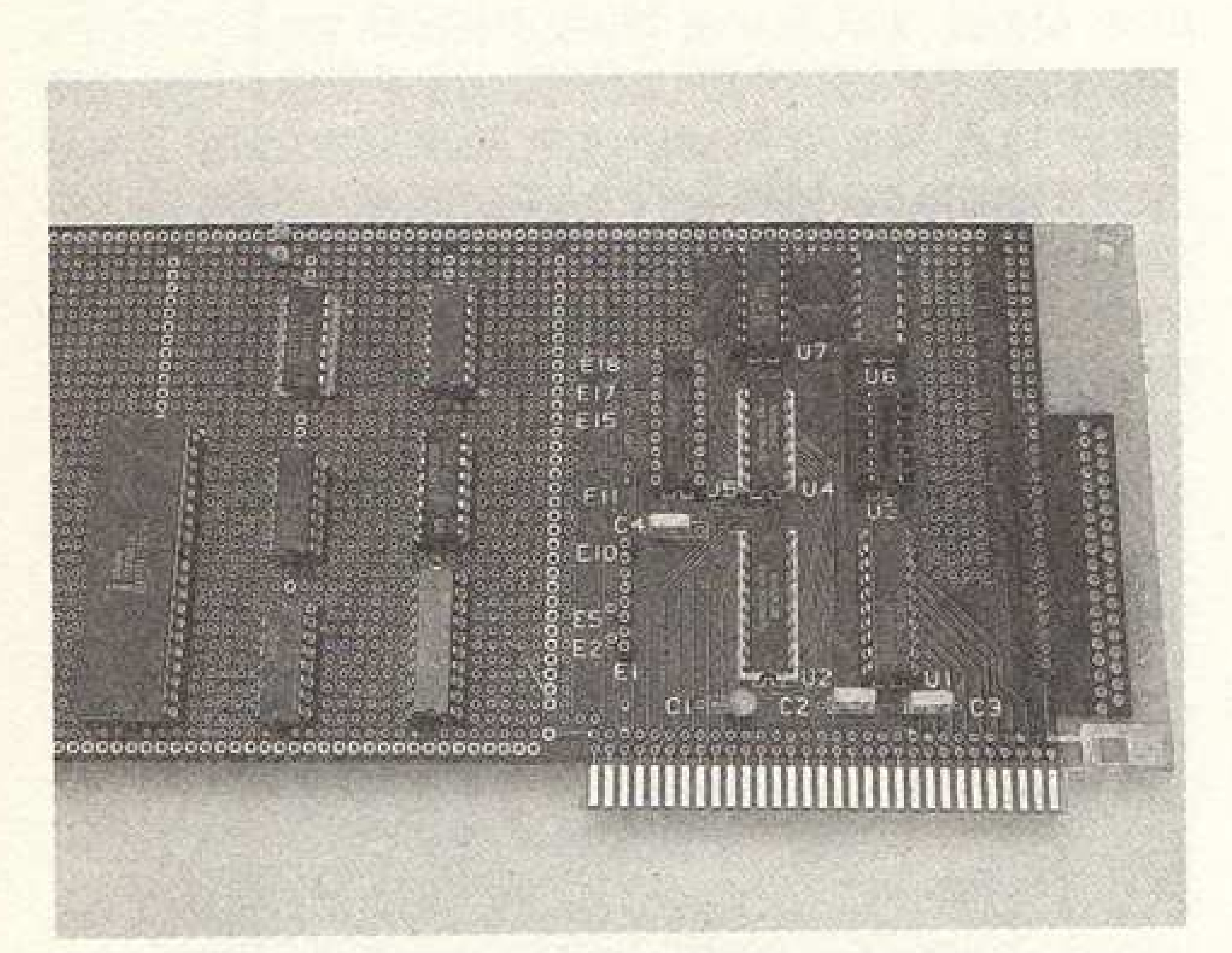


Fig. 6. Montaje mediante hilo enrollado sobre tarjeta de prototipos IBM-PC.

El montaje mediante hilo enrollado resulta de gran robustez y fiabilidad, siendo bastante caro frente a otras soluciones de tipo aficionado. Presenta otra ventaja adicional y es que resulta bastante fácil recablear un circuito para reparar un error o añadir un circuito. Se desenrollan los pines afectados y se recablean para realizar el nuevo circuito.

Solución sobre circuito impreso

Si el número de circuitos a realizar es numeroso, la solución más económica es colocar todos las conexiones posibles sobre el mismo soporte.

Para ello se emplean circuitos impresos de 1, 2 o más capas, según las necesidades.

El diseño con circuito impreso implica todos los pasos indicados para los procedimientos anteriores y además el diseño físico de la distribución de pistas y su representación gráfica con la calidad adecuada para su posterior representación mediante técnicas fotográficas o serigráficas sobre un soporte metálico que pueda trabajarse física o químicamente. Es cada vez más frecuente el encontrar aficionados que son capaces de generar sus propios circuitos y procesarlos químicamente hasta doble cara. Existen programas para ordenadores personales que ayudan en la preparación de los circuitos impresos. Mostramos un ejemplo del circuito de ampliación de puertos de E/S realizado para dos caras. Los contactos de la capa superior deberán hacerse mediante puentes. Para hacerlo a doble cara y taladro metalizado es necesario disponer de equipo muy costoso y que no se justifica para trabajos de poco volumen.

No es necesario recalcar mucho que la solución idónea para añadir equipos periféricos a nuestro ordenador personal es mediante circuito impreso.

El esquema de la figura ha sido realizado con la impresora de tipo matricial de un ordenador personal, mediante un programa de diseño gráfico.

La impresora genera la plantilla a doble tamaño del original, realizando varias pasadas para conseguir suficiente contraste. Es necesario reducir fotográficamente el dibujo para obtener el negativo que por contacto permitirá generar el circuito impreso. El circuito es funcionalmente igual al realizado mediante circuito de tiras. Es necesario completar el montaje con la conexión de las señales de control de los puertos desde los circuitos de decodificación, mediante hilo aislado.

Ensayos de la tarjeta

Repasaremos todas las conexiones comprobando la continuidad de los circuitos, en-

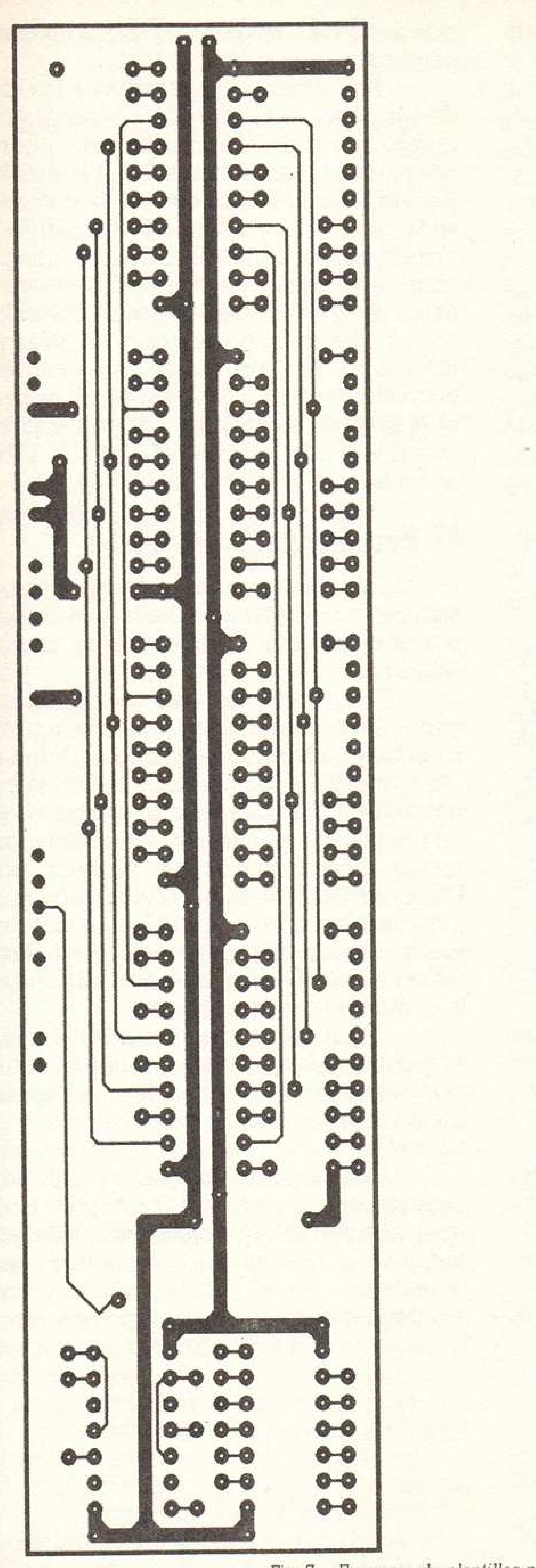
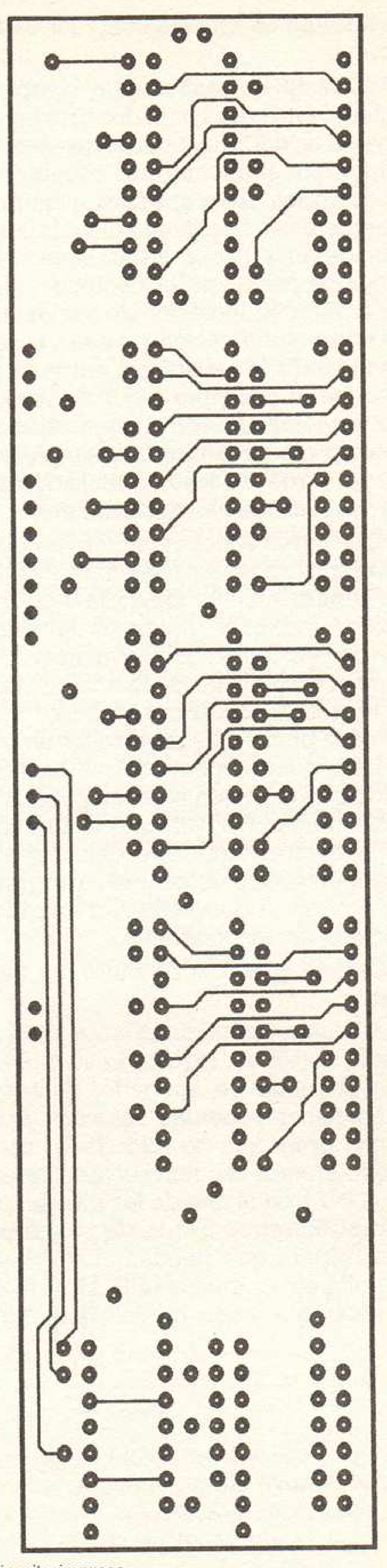



Fig. 7. Esquema de plantillas para circuito impreso.

tre los terminales del conector y las patillas de los zócalos.

Es también conveniente comprobar el aislamiento entre los contactos próximos, pues a veces las soldaduras pueden presentar hilos muy finos que al encontrarse recubiertos con restos de resina no se aprecian a simple vista. Igualmente entre la alimentación de +5 voltios y masa debería haber aislamiento completo hasta que se coloquen los componentes. Si se coloca algún condensador de varios microfaradios entre alimentación y masa, se observará que al medir la resistencia entre +5 y masa parece que al principio hay baja resistencia, que va aumentando al principio rápidamente y luego con mayor lentitud. Es simplemente el efecto de carga de los condensadores a través de las resistencias del polímetro.

Colocaremos los circuitos del decodificador de direcciones y conectaremos la tarjeta, con la alimentación apagada. Los circuitos integrados son muy delicados en cuanto al sentido de la orientación. Normalmente pagan con la muerte súbita la aplicación de la alimentación en sentido contrario. Algunos circuitos de potencia pueden llegar a soltar un poco de humo, pero generalmente sufren calladamente y a lo sumo se calientan a temperaturas fuera de lo habitual. Aplicaremos la alimentación y comprobaremos que el ordenador arranca correctamente. Con un pequeño programa podemos activar el decodificador produciendo la dirección correspondiente.

Deberá aparecer un pulso por cada activación.

A continuación añadiremos los circuitos de puerto de salida, siguiendo siempre la secuencia obligada de apagado, desconexión, inserción de componentes, conexión, encendido. Con el programa de tests de las direcciones produciremos secuencias de niveles estables, con 0 y 1 cada uno de los bits del puerto. Luego produciremos trenes de pulsos de baja frecuencia para que puedan observarse mediante polímetro o diodos LED. Es el momento de verificar que cada bit sale por donde le corresponde, es decir, que no hemos intercambiado ningún hilo.

Por último, completaremos los circuitos de los puertos de entrada. La verificación requiere la exploración continua del puerto, con presentación en pantalla del resultado de la lectura, mientras activamos las entradas con señales 0 y 1, producidas mediante la conexión de una resistencia de, por ejemplo, 220 ohms, a +5 voltios y a masa. Al conectar las señales de prueba a las entradas deberá tenerse la precaución de no tocar en otras patillas diferentes. Si la prueba se hace con resistencias de 220 ohms o superiores, no es probable que se dañe el circuito, pero es muy posible que si el punto tocado es del bus, el programa se vaya por los cerros de Ubeda.

Recomendaciones finales

Es recomendable extremar las precauciones, para evitar que haya que lamentarse por una conexión incorrecta o un circuito colocado al revés.

Colocar en los conectores alguna plaquita de orientación para que sea imposible la inserción en sentido contrario al debido.

Verificar que la línea de masa está conectada en todas las tarjetas utilizadas, y la alimentación es la interna del ordenador solamente si los circuitos son de bajo consumo. Disponer de otra alimentación suplementaria para la conexión de periféricos de consumo elevado como motores, relés y similares. La alimentación deberá estar limitada en corriente y con fusible de protección.

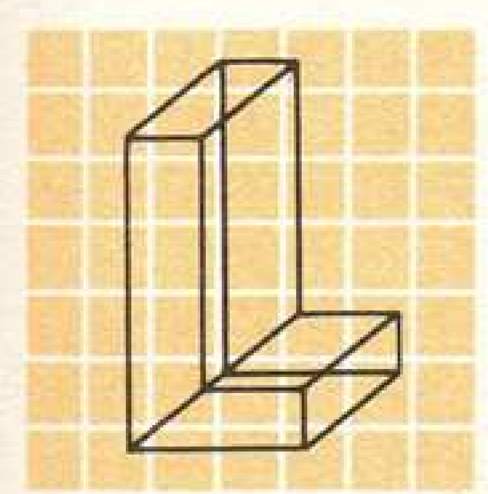
Utilizar siempre circuitos de aislamiento mediante fotoacoplador o relé si el dispositivo de carga se conecta a la red. Lo mismo es aplicable si las señales a medir son de muy bajo nivel.

Con algunos pequeños instrumentos montados por nosotros mismos, podemos verificar cómo opera la tarjeta hasta los más finos detalles. Lo iremos viendo a medida que nos adentremos en la práctica de la circuitería asociada a nuestro ordenador personal.

APRENDER CON EL ORDENADOR

NATURALEZA Y TECNOLOGIA

Anatomía humana. Los huesos



OS huesos constituyen el sistema fundamental de sostén del organismo, así como los elementos pasivos del movimiento. Son cada una de las piezas duras y resistentes, formadas por sustancia orgánica y sales minerales, y envueltas por una membrana fibrosa que forman nuestro esqueleto.

En relación con su forma podemos distinguir tres tipos:

1) Huesos largos. Formados por una caña de hueso (llamada diáfisis) y dos extremos (llamados epífisis). En su interior existe una cavidad central o conducto medular. Un ejemplo es el fémur.

2) Huesos anchos. Constituidos por dos láminas de tejido óseo compacto entre las que se interpone tejido óseo esponjoso. Un ejemplo son los huesos de la bóveda del cráneo.

3) Huesos cortos. Sus tres dimensiones son casi iguales. Están formados por tejido esponjoso revestido de una débil capa de tejido óseo compacto.

La membrana fibrosa que reviste los huesos se llama periostio y es la que confiere vitalidad al hueso.

Los huesos se engranan entre sí por medio de articulaciones, y las superficies óseas en contacto están recubiertas de cartílago; la cápsula articular es un manguito fibroso que rodea a la articulación, que a su vez está revestida interiormente por la membrana sinovial. El líquido sinovial llena la articulación y hace la función de lubricante en el roce de las superficies óseas articulares. La cápsula queda reforzada por los ligamentos.

Una vez conocido el fundamento del aparato locomotor, creemos que te resultará interesante el del conocimiento de sus traumatismos. Posiblemente te haya intrigado el ver a algún amigo con una escayola, ¿Qué le ha sucedido?, ¿cómo lo notó?:

— Si tu amigo ha sufrido un esguince o torcedura ha sido porque debido a un movimiento forzado una articulación se ha abierto más de sus límites fisiológicos. Las superficies articulares pierden su contacto sólo de modo pasajero y una vez desaparecida la fuerza que ha actuado, recobra su posición primitiva, quedando lesionada alguna parte blanda.

Al ocurrirle sintió dolor, se le inflamó la articulación por el derrame, le apareció un cardenal y tenía dificultad para realizar el movimiento.

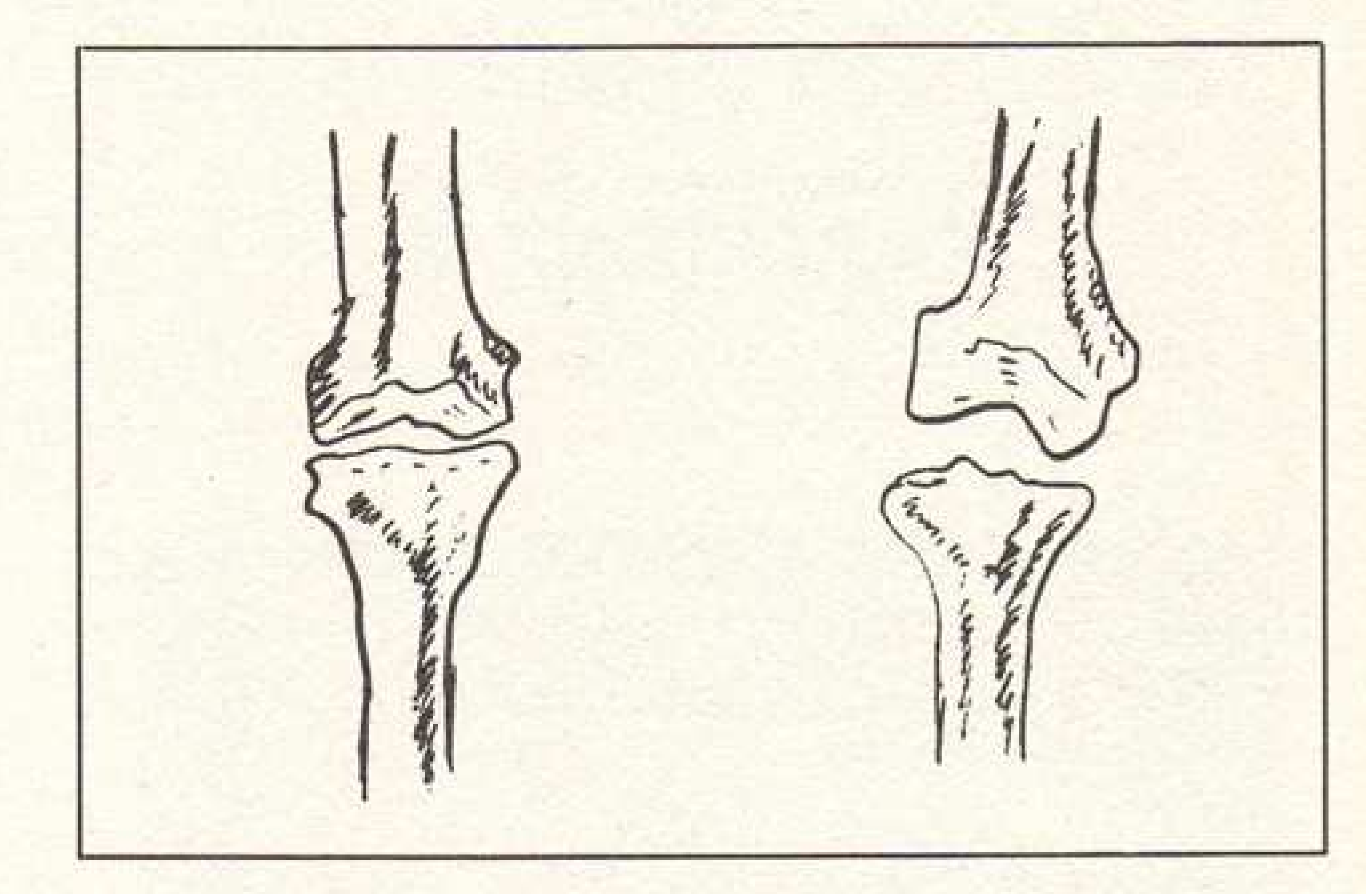


Fig. 1. Ejemplo de esguince.

— Si tu amigo ha sufrido una luxación ha sido por la salida de los huesos de su sitio, pero, a diferencia del esguince, esta separación es de modo permanente.

Los síntomas que tuvo fueron una gran

deformidad, un dolor intenso al presionar y la imposibilidad del movimiento.

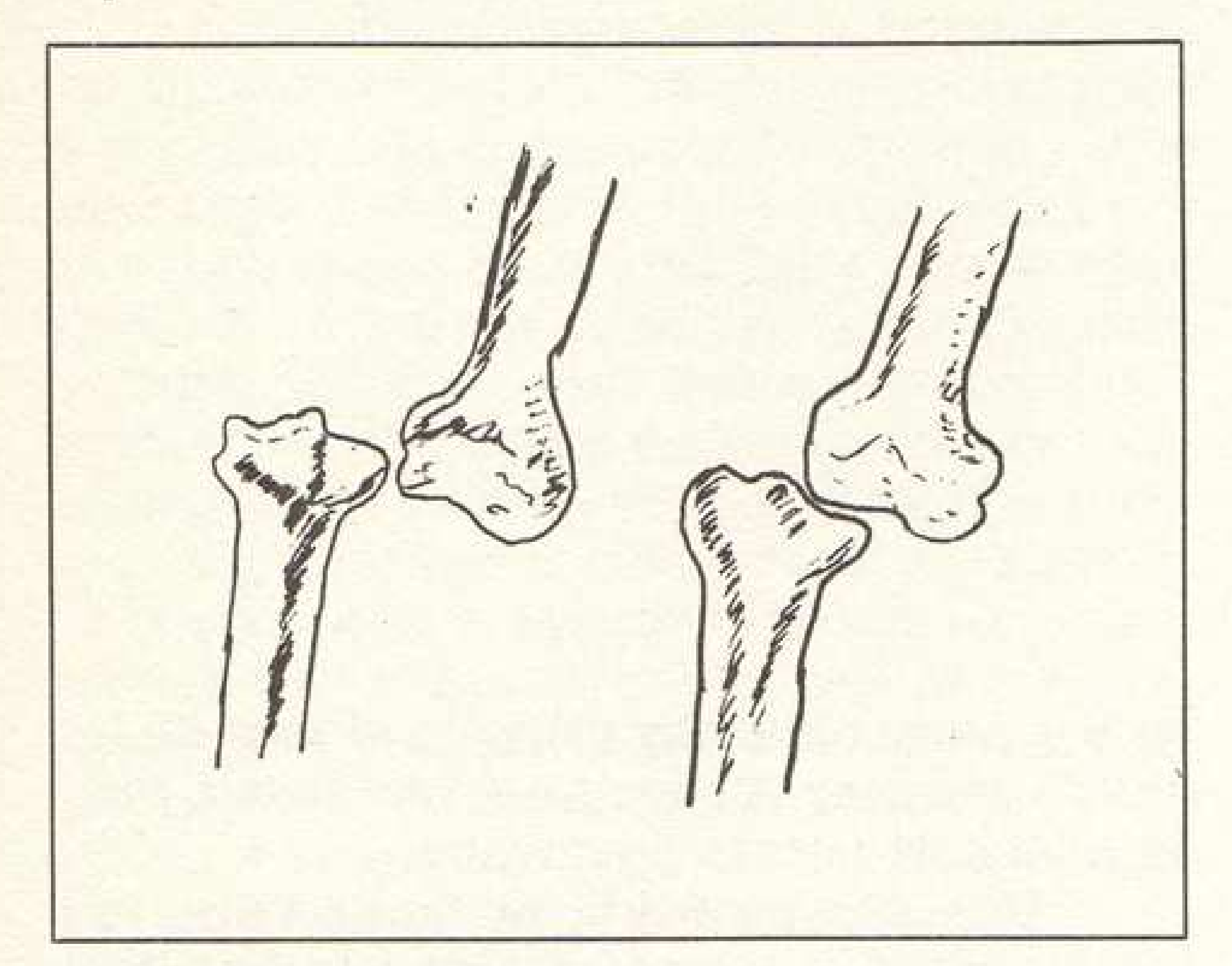


Fig. 2. Ejemplo de luxación.

— Si tu amigo ha sufrido una fractura, se le ha roto el hueso, total o parcialmente. Hay muchos tipos de fracturas, en general todas son muy dolorosas.

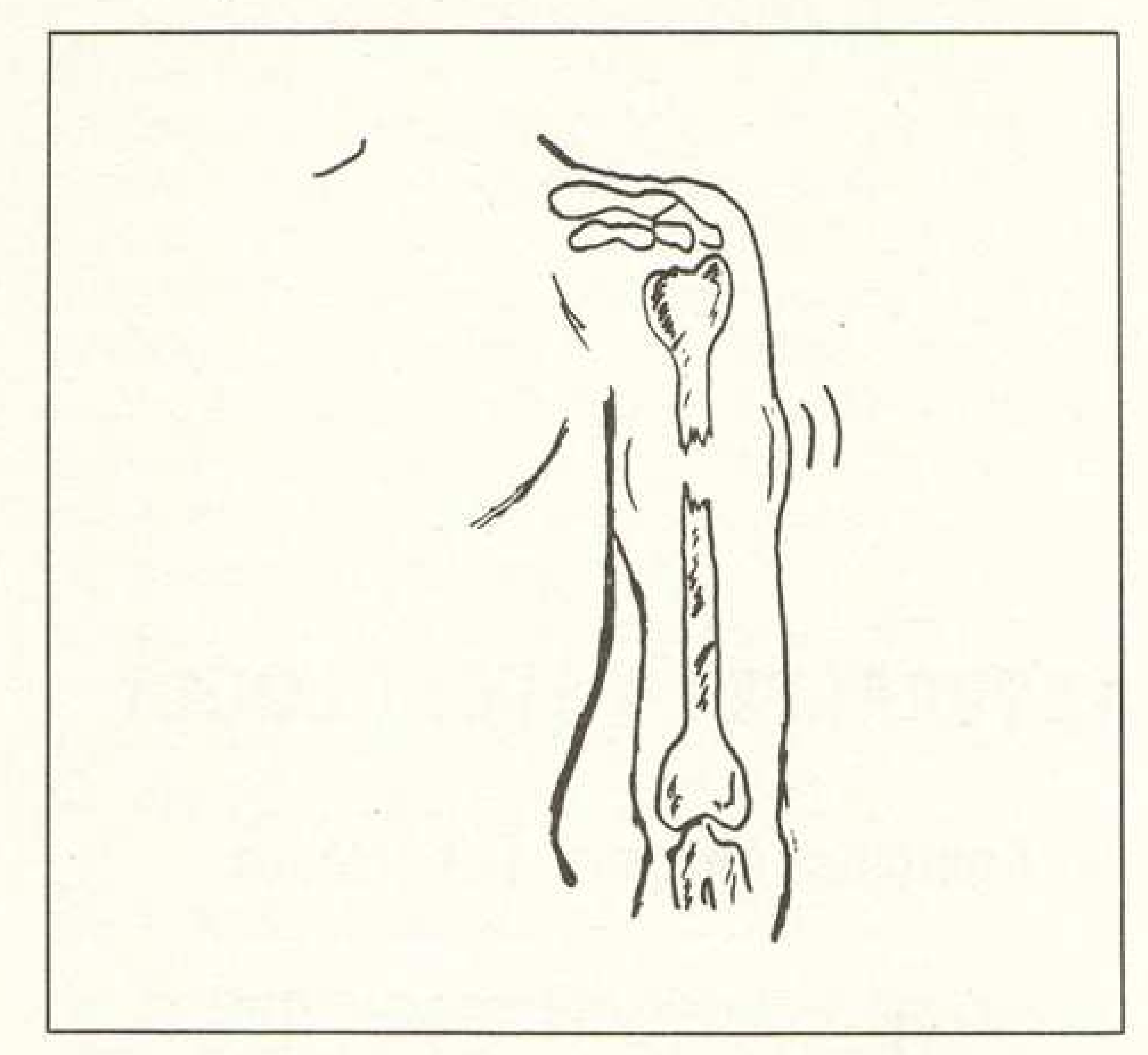


Fig. 3. Ejemplo de fractura.

Ante cualquiera de estos casos hay que acudir al médico, ya que es importante que los huesos queden bien, son nuestro aparato locomotor. ¡Cuídalo!.

El ordenador te va a ayudar a recordar algunos nombres importantes de huesos del aparato locomotor.

El funcionamiento de este programa es muy sencillo; es un test al que debes responder con el número de la opción correcta.

```
10 REM *********************************
20 REM ** PROGRAMA DE ANATOMIA HUMANA **
30 REM **       PARA ZX-SPECTRUM      **
40 REM *********************************
50 RESTORE: FLASH 0: CLS: POKE 23609,255
60 BORDER 1: PAPER 1: INK 6
70 DIM N$(20,10): DIM N(20,2)
80 FOR I=1 TO 20
90 READ N(I,1),N(I,2),N$(I)
100 NEXT I
110 REM ***********************
120 REM **   DIBUJO CABEZA   **
130 REM ***********************
140 PLOT 16,148
150 DRAW 12,9,-4.18
160 FOR I=1 TO 6
170 READ A,B
180 DRAW A,B
190 NEXT I
200 CIRCLE 25,152,1
210 REM ***********************
220 REM **   DIBUJO TRONCO   **
230 REM ***********************
240 PLOT 16,148
250 FOR I=1 TO 9
260 READ A,B
270 DRAW A,B
280 NEXT I
290 PLOT 31,100: DRAW 1,-4: DRAW 4,-1
300 PLOT 31,100: DRAW -1,-4: DRAW 1,-2
310 PLOT 31,100
320 FOR I=1 TO 22
330 READ A,B
340 DRAW A,B
350 NEXT I
```

```
360 PLOT 19,147
370 FOR I=1 TO 4
380 READ A,B
390 DRAW A,B
400 NEXT I
410 REM ******************
420 REM **   COSTILLAS   **
430 REM ******************
440 PLOT 21,31: DRAW 5,-2
450 PLOT 22,125: DRAW 5,-2
460 PLOT 22,122: DRAW 5,-2
470 PLOT 23,119: DRAW 5,-2
480 PLOT 21,128: DRAW 5,-2
490 PLOT 23,112: DRAW -6,19
500 PLOT 12,47
510 REM ***************
520 REM **   PIERNA   **
530 REM ***************
540 FOR I=1 TO 11
550 READ A,B
560 DRAW A,B
570 NEXT I
580 REM ************************************
590 REM **   PONER PUNTOS PARA PREGUNTAS   **
600 REM ************************************
610 FOR X=1 TO 20
620 CIRCLE INK 2: N(X,1),N(X,2),2
630 LET P= INT(RND*3)+1
640 LET H=X
650 REM *************************
660 REM **    PONER PREGUNTAS   **
670 REM *************************
680 FOR I=1 TO 3
690 PRINT AT I*2+5,14;I
700 PRINT AT P*2+5,16;"-";N$(H)
710 LET P=P+1: IF P=4 THEN LET P=1
720 LET H=H+1: IF H>20 THEN LET H=H-5
730 NEXT I
740 INPUT "CUAL ES LA CORRECTA?";D
750 IF D=P THEN PRINT AT 20,12;FLASH 1;"       CORRECTO       ":FOR Z=1 TO 5: BEEP
    .15,Z: NEXT Z: GOTO 770
760 PRINT AT 20,12;  "   INCORRECTO   ":FOR Z=1 TO 100: NEXT Z: LET X=X-1: GOTO
    780
770 CIRCLE INK 7 : N(X,1),N(X,2),2
780 NEXT X
790 FLASH 1
800 PRINT "QUIERES INTENTARLO OTRA VEZ?"
810 LET V$=INKEY$: IF V$="" THEN GOTO 810
820 IF V$="S" OR V$="s" THEN GOTO 50
830 GOTO 9999
840 DATA 25,159,"FRONTAL",14,159,"PARIETAL",14,151,"OCCIPITAL",12,130,"HOMOPLAT
         O",24,126,"COSTILLAS",27,145,"MAXILARES",18,121,"HUMERO",17,142,"CERVI
         CALES",25,106,"CUBITO",19,95,"ILIACO"
850 DATA 12,9,"TARSO",26,131,"ESTERNON",16,136,"CLAVIVULA",13,86,"COXIS",28,150
         ,"NASAL",34,97,"FALANGES",14,25,"PERONE",17,46,"TIBIA",19,71,"FEMUR",1
         9,101,"VERTEBRAS"
860 DATA 1,-5,-1,-3,1,-3,-1,-4,-8,2,1,5
870 DATA -1,-9,-5,-7,1,-12,5,8,-1,2,4,-19,4,1,8,-12,5,-2
880 DATA -12,11,2,-15,6,-4,-4,-5,4,-4,-6,-5,-1,-18,-3,-12,-4,1,5,17,-2,15,-1,0,
         0,3,3,6,-4,0,-1,-4,-1,6,4,3
890 DATA 5,0,-4,0,1,10,-5,14,-1,-8,8,-8,4,-12,-3,-4
900 DÁTA 6,2,-3,-36,6,-3,6,-2,-19,-1,3,6,4,0,0,31,-3,3,0,-25,-1,-9,25,159
```

Programa 1. Test de huesos (versión Spectrum).

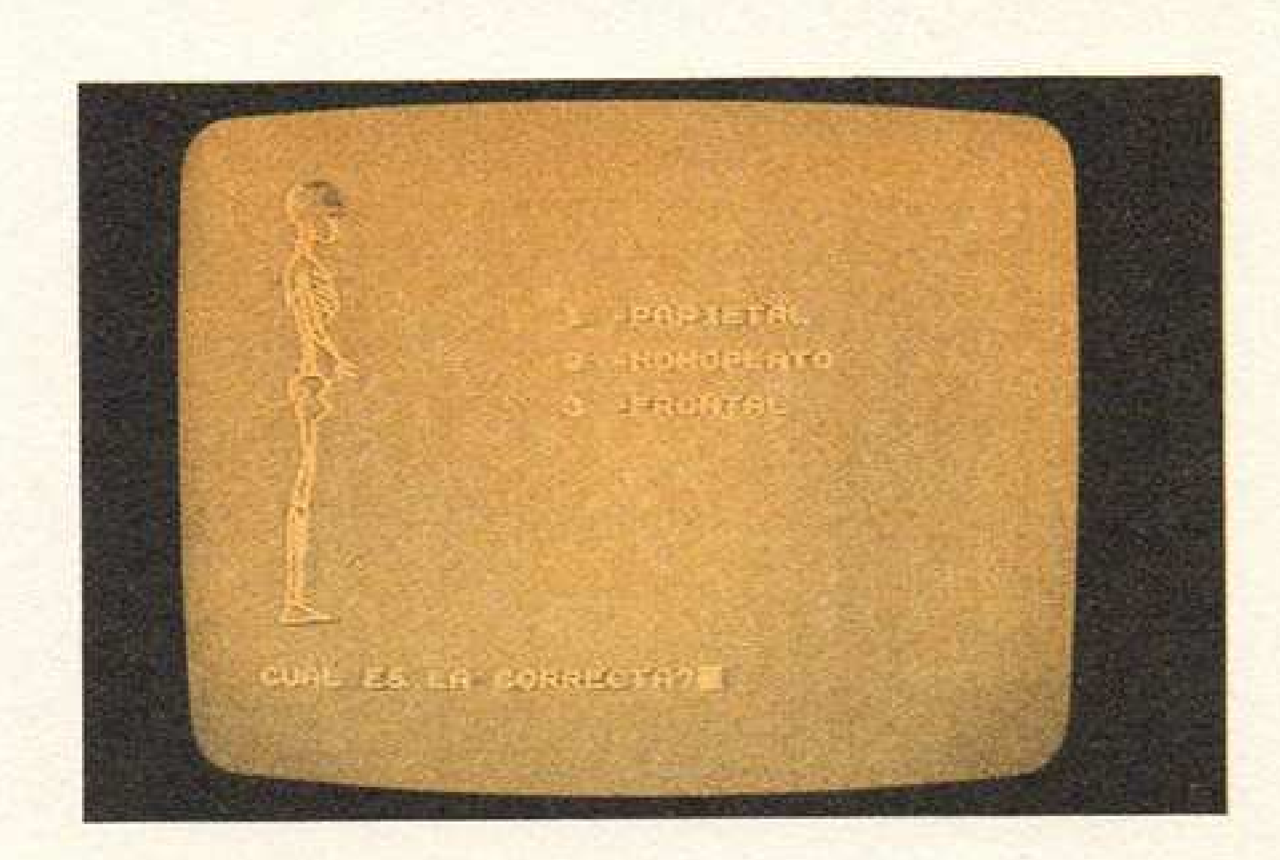


Foto 1. Se trata de descubrir el hueso frontal.

El programa consta de dos partes; en la primera se realiza el dibujo del esqueleto humano, en la segunda se realiza el test propiamente dicho.

Modificaciones para otros equipos

Vamos a ir viendo qué órdenes es necesario que varíes para adaptar este programa a tu ordenador.

Elimina las órdenes FLASH, POKE, INK, BORDER, PAPER y BEEP que encuentres. Se utilizan para introducir características atractivas al programa, que puedes hacer luego a tu gusto: colores, música, parpadeos, etc.

Ahora viene lo más difícil, el gráfico del esqueleto:

1) Los PLOT sitúan un punto en la pantalla y los DRAW trazan una línea de forma relativa, es decir, aumentan la coordenada en el valor especificado.

```
PLOT 3,4        3,4*----------
                              |
                              *15,1
```

2) PLOT y DRAW con AMSTRAD. Funcionan igual si utilizas DRAWR (DRAWR 5,-3).

Si utilizas el DRAW x,y tienes que dar coordenadas absolutas (DRAW 15,1 en el ejemplo).

3) PLOT y DRAW en MSX. Funcionan del mismo modo si utilizas signos + y - al indicar las coordenadas, es decir, en nuestro ejemplo sería: DRAW +5,-3. Si omites los signos, lo considerará como coordenadas absolutas.

4) COMMODORE. Dependiendo de la versión que tengas, es posible que no dispongas de estas órdenes en el repertorio. En ese caso tendrás que utilizar una sentencia SPRITE o realizarlo mediante POKE.

5) PLOT y DRAW en IBM-PC. No dispone de PLOT, pero hay posibilidad de situar un punto en la pantalla con el DRAW si se le añade el comando B delante de la coordenada, para indicar que desplace sin dibujar la línea.

Por otra parte, si las coordenadas las das en modo relativo, no olvides los signos + y - al igual que en MSX.

Otras sentencias que aparecen y debes cambiar son las que tienen PRINT AT:

1) En ZX-SPECTRUM es PRINT AT línea, columna.

2) En AMSTRAD hay que situar primero el cursor y luego se utiliza un PRINT.

Será LOCATE columna, línea, por lo que debes invertir el orden de los argumentos en el programa.

3) MSX también utiliza el LOCATE columna, línea.

4) COMMODORE no tiene sentencia para localizar el cursor en todas las versiones; por tanto, debes recurrir si es tu caso a:

POKE 1024+columna+40*fila.

5) IBM utiliza LOCATE línea, columna.

Por último, veamos la generacion de números aleatorios.

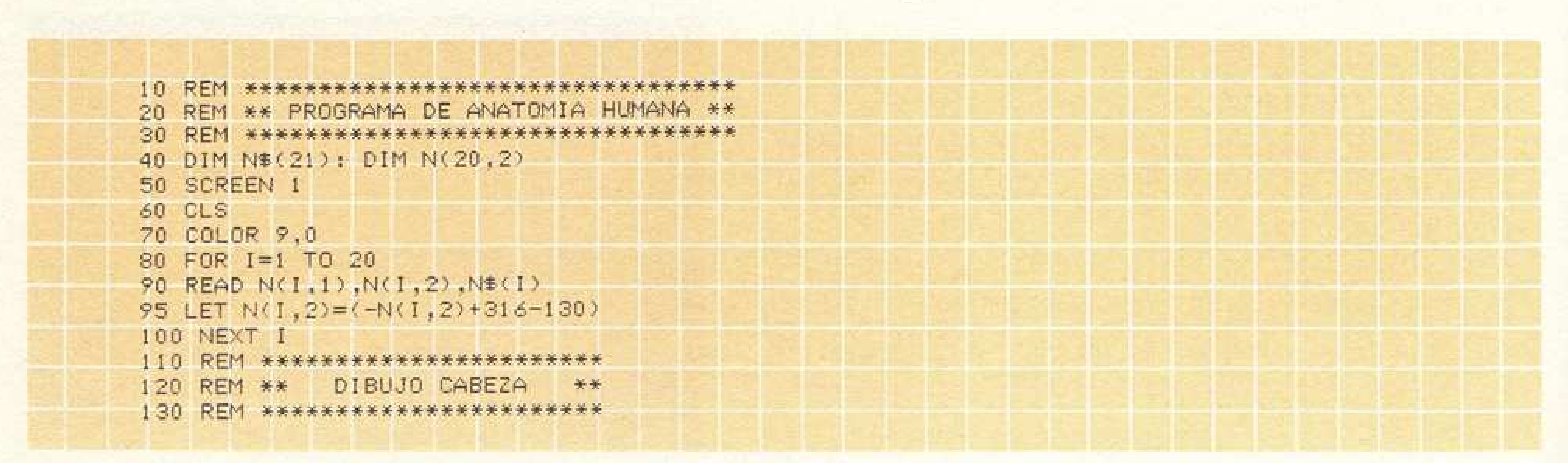
AMSTRAD utiliza para generar la secuencia aleatoria un RANDOMIZE. Te aconsejamos utilizar el RANDOMIZE TIME.

MSX y COMMODORE añaden un parámetro para la generación de números. Sustituye la sentencia por:

LET P= INT (RND(1)*3)+1

IBM necesita un RANDOMIZE TIME o RANDOMIZE n, siendo n un número entre -32767 y 32768.

Queda recordarte que en el ZX-SPECTRUM al dimensionar una matriz alfanumérica es necesario darle la longitud, cosa que no ocurre en los demás casos. Y que termina el programa con un GOTO 9999, que será sustituido por un END.



```
10 REM **********************************
20 REM ** PROGRAMA DE ANATOMIA HUMANA **
30 REM **********************************
40 DIM N$(21): DIM N(20,2)
50 SCREEN 1
60 CLS
70 COLOR 9,0
80 FOR I=1 TO 20
90 READ N(I,1),N(I,2),N$(I)
95 LET N(I,2)=(-N(I,2)+316-130)
100 NEXT I
110 REM ***********************
120 REM **   DIBUJO CABEZA    **
130 REM ***********************
```

```
140 PSET (16, 40)
160 FOR I=1 TO 10
170 READ A$,B :LET B=B*(-1) :LET B$=STR$(B)
175 LET DIB$="M"+A$+","+B$
180 DRAW DIB$
190 NEXT I
210 REM ***********************
220 REM **   DIBUJO TRONCO   **
230 REM ***********************
240 PSET (16, 40)
250 FOR I=1 TO 9
260 READ A$,B: LET B=B*(-1): LET B$=STR$(B)
265 LET DIB$="M"+A$+","+B$
270 DRAW DIB$
280 NEXT I
290 PSET (31,88): DRAW "M+1,+4": DRAW "M+4,+1"
300 PSET (31,88): DRAW "M-1,+4": DRAW "M+1,+2"
310 PSET (31,88)
320 FOR I=1 TO 22
330 READ A$,B: LET B=B*(-1): LET B$=STR$(B)
335 LET DIB$="M"+A$+","+B$
340 DRAW DIB$
350 NEXT I
360 PSET (19,41)
370 FOR I=1 TO 4
380 READ A$,B: LET B=B*(-1): LET B$=STR$(B)
385 LET DIB$="M"+A$+","+B$
390 DRAW DIB$
400 NEXT I
410 REM ******************
420 REM **   COSTILLAS   **
430 REM ******************
440 PSET (21,57): DRAW "M+5,+2"
450 PSET (22,63): DRAW "M+5,+2"
460 PSET (22,66): DRAW "M+5,+2"
470 PSET (23,69): DRAW "M+5,+2"
480 PSET (21,60): DRAW "M+5,+2"
490 PSET (23,76): DRAW "M-6,-19"
500 PSET (12,141)
510 REM ****************
520 REM **   PIERNA   **
530 REM ****************
540 FOR I=1 TO 11
550 READ A$,B: LET B=B*(-1): LET B$=STR$(B)
555 LET DIB$="M"+A$+","+B$
560 DRAW DIB$
570 NEXT I
580 REM ************************************
590 REM **   PONER PUNTOS PARA PREGUNTAS   **
600 REM ************************************
610 FOR X=1 TO 20
620 CIRCLE (N(X,1),N(X,2)),1,15
630 LET P= INT(RND*3)+1
640 LET H=X
650 REM *************************
660 REM **   PONER PREGUNTAS   **
670 REM *************************
680 FOR I=1 TO 3
690 LOCATE I*2+5,25:PRINT I
700 LOCATE P*2+5,27:PRINT "-";N$(H)
710 LET P=P+1: IF P=4 THEN LET P=1
720 LET H=H+1: IF H>=20 THEN LET H=H-10
730 NEXT I
740 LOCATE 20,13:INPUT "CUAL ES LA CORRECTA?   ";D
750 IF D=P THEN LOCATE 20,12:PRINT "      CORRECTO                  ":FOR Z=1 TO 10
00:NEXT Z: GOTO 770
760 LOCATE 20,10:PRINT  "        INCORRECTO                ":FOR Z=1 TO 500: NEXT Z:
    LET X=X-1:GOTO 780
770 CIRCLE (N(X,1),N(X,2)),1,0
780 NEXT X
800 PRINT "QUIERES INTENTARLO OTRA VEZ?"
810 LET V$=INKEY$: IF V$="" THEN GOTO 810
820 IF V$="S" OR V$="s" THEN GOTO 50
830 END
840 DATA 25,159,"FRONTAL   ",14,159,"PARIETAL  ",14,151,"OCCIPITAL  ",12,130,"HO
MOPLATO  ",24,126,"COSTILLAS  ",27,145,"MAXILARES  ",18,121,"HUMERO     ",17,142
,"CERVICALES ",25,106,"CUBITO    ",19,95,"ILIACO     "
850 DATA 12,9,"TARSO      ",26,131,"ESTERNON   ",16,136,"CLAVICULA   ",13,86,"CO
XIS     ",28,150,"NASAL       ",34,97,"FALANGES     ",14,25,"PERONE      ",17,46
,"TIBIA     ",19,71,"FEMUR     ",19,101,"VERTEBRAS     "
859 DATA -1,+15,+5,+4,+6,+0,+4,-3
```

```
860 DATA +1,-5,-1,-3,+1,-3,-1,-4,-8,+2,+1,+5
870 DATA -1,-9,-5,-7,+1,-12,+5,+8,-1,+2,+4,-19,+4,+1,+8,-12,+5,-2
880 DATA -12,+11,+2,-15,+6,-4,-4,-5,+4,-4,-6,-5,-1,-18,-3,-12,-4,+1,+5,+17,-2,
         +15,-1,+0,+0,+3,+3,+6,-4,+0,-1,-4,-1,+6,+4,+3
890 DATA +5,+0,-4,+0,+1,+10,-5,+14,-1,-8,+8,-8,+4,-12,-3,-4
900 DATA +6,+2,-3,-36,+6,-3,+6,-2,-19,-1,+3,+6,+4,+0,+0,+31,-3,+3,+0,-25,-1,-9,+
25,+159
```

Versión para IBM del programa 1.

MATEMATICAS

Programa de números complejos

Un número complejo es un número que consta de dos partes:

— Una parte real.
— Otra imaginaria.

Una posible forma de interpretar esta descomposición es considerar el número complejo como un vector, cuya parte real es su proyección ortogonal en el eje X y la parte imaginaria su proyección en el eje Y.

La letra i es la reprentación de $\sqrt{-1}$.
Para ella tendremos:

$$i = \sqrt{-1}$$

$$i^2 = -1$$

$$i^3 = -i$$

$$i^4 = 1$$

A partir de la cuarta potenciación esta secuencia se repite.

Operaciones con números complejos

Sumar y restar números complejos es sumamente sencillo; basta operar por separado las partes real e imaginaria.

Sean dos números complejos:

$$Z = a + bi$$
$$W = c + di$$

Siendo a,b,c,d números cualesquiera (positivos o negativos).

$$Z + W = (a + c) + (b + d)i$$
$$Z - W = (a - c) + (b - d)i$$

El producto es más complicado, pues al multiplicar i por i aparecerán potencias con las que hay que operar cuidadosamente.

$$Z^*W = (a + bi)(c + di) =$$
$$= ac + adi + bci + bdi^2 = (ac - bd) + (ad + bc)\,i$$

La división obliga a introducir el concepto de conjugado de un número complejo:

Sea $W = a + bi$

Su conjugado (que se representa por W) es:

$$W = a - bi$$

Pues bien, para dividir dos números complejos hay que multiplicar por el conjugado:

$$\frac{a + bi}{c + di} = \frac{(a + bi)(c - di)}{(c + di)(c - di)} =$$

$$= \frac{(a + bi)(c - di)}{(c^2 + d^2)} = \frac{ac - adi\ bci - bdi^2}{(c^2 + d^2)} =$$

$$= \frac{ac + bd}{(c^2 + d^2)} + \frac{bc - ad}{(c^2 + d^2)}\, i$$

El programa

El programa realiza las operaciones de suma, resta, multiplicación y división de dos números complejos. Visualiza en pantalla la representación gráfica, tanto de los operandos como el resultado. Para esta representación realiza un cambio de escala para adaptarla al valor más grande en cada caso.

El programa pide como datos:

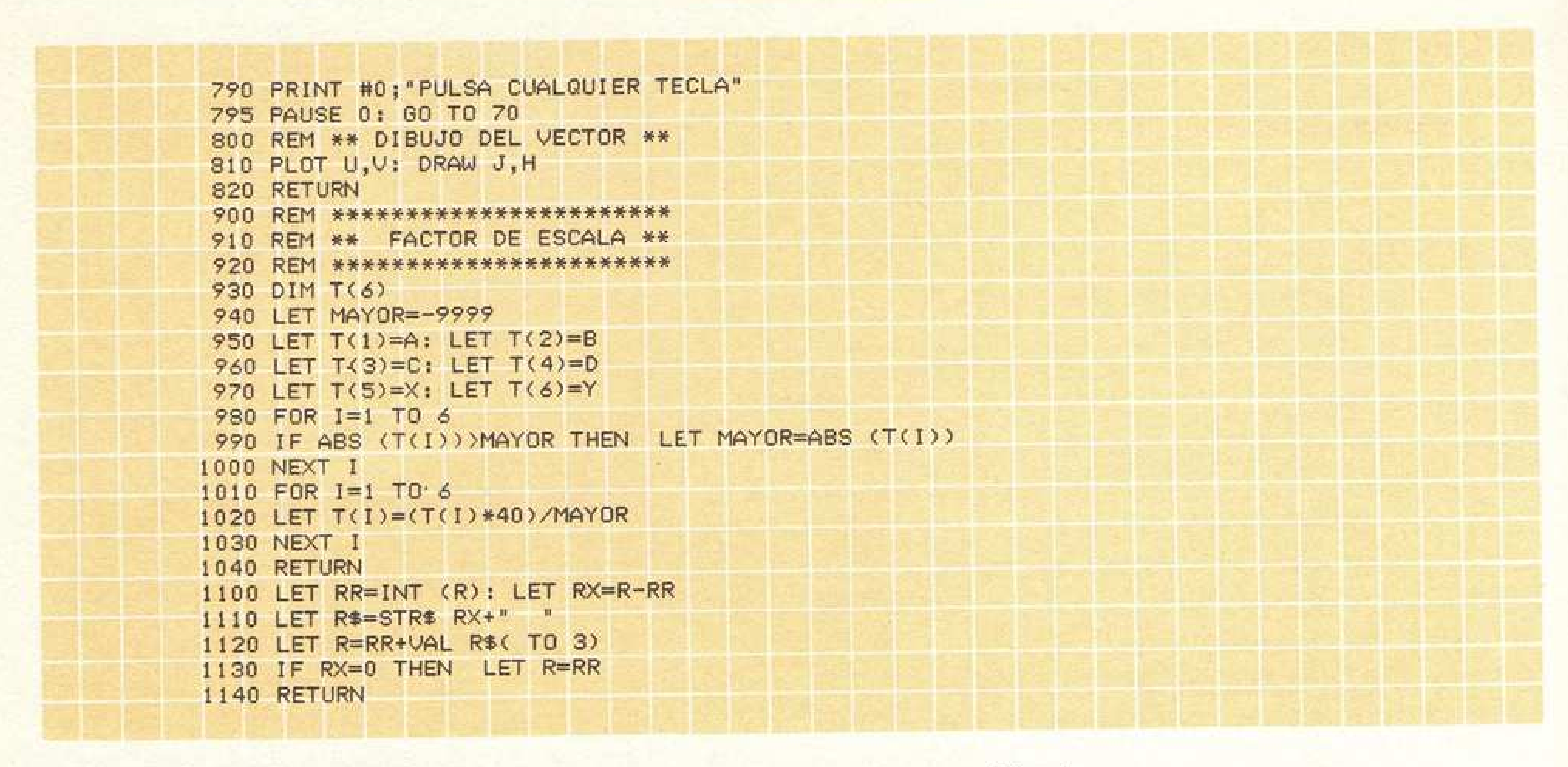
— La operación a realizar (+,-,*, \).
— Cada uno de los componentes de los dos operandos.

El programa acepta cualquier número de decimales y opera con ellos; pero al imprimirlos en pantalla los redondea a un único de-

cimal para evitar número indefinido de decimales que podían aparecer al operar.

Nota: Como tanto los operandos como el resultado se representan en la misma escala para dar una imagen gráfica más real, si alguno de los valores es muy grande puede hacer que la representación de los otros sea extremadamente pequeña.

```
 10 REM ********************
 20 REM **NUMEROS COMPLEJOS**
 30 REM ********************
 40 PRINT "********************************"
 50 PRINT "****   NUMEROS  COMPLEJOS   ****"
 60 PRINT "********************************"
 70 PAUSE 100: CLS
 80 PRINT "INTRODUCE LOS DATOS            ";
 90 INPUT "OPERACION QUE QUIERE REALIZAR:  (+,-,*,/)",X$
100 IF (X$<>"+" AND X$<>"-" AND X$<>"*" AND X$<>"/") THEN  GO TO 70
110 INPUT "PARTE REAL DEL PRIMER NUMERO",A
120 INPUT "PARTE IMAGINARIA DEL PRIMER      NUMERO",B
130 INPUT "PARTE REAL DEL SEGUNDO  NUMERO",C
140 INPUT "PARTE IMAGINARIA DEL SEGUNDO     NUMERO",D: CLS
150 IF X$="+" THEN  GO TO 200
160 IF X$="-" THEN  GO TO 300
170 IF X$="*" THEN  GO TO 400
180 IF X$="/" THEN  GO TO 500
200 REM **********************
210 REM ****     SUMA      ****
220 REM **********************
230 LET X=A+C
240 LET Y=B+D
250 GO TO 550
300 REM **********************
310 REM ****     RESTA     ****
320 REM **********************
330 LET X=A-C
340 LET Y=B-D
350 GO TO 550
400 REM **********************
410 REM *** MULTIPLICACION  ***
420 REM **********************
430 LET X=A*C-B*D
440 LET Y=A*D+B*C
450 GO TO 550
500 REM **********************
510 REM ***    DIVISION     ***
520 REM **********************
530 LET X=(A*C+B*D)/(C*C+D*D)
540 LET Y=(B*C-A*D)/(C*C+D*D)
550 GO SUB 900
560 REM **********************
570 REM REPRESENTACION  GRAFICA
580 REM **********************
590 REM ** DIBUJO DE LOS EJES**
600 LET R=A: GO SUB 1100: LET A=R
610 LET R=B: GO SUB 1100: LET B=R
620 LET R=C: GO SUB 1100: LET C=R
625 LET R=D: GO SUB 1100: LET D=R
630 LET R=X: GO SUB 1100: LET X=R
635 LET R=Y: GO SUB 1100: LET Y=R
640 PLOT 24,128: DRAW 80,0
650 PLOT 63,88: DRAW 0,80
652 LET B$="+": IF B<0 THEN  LET B$=""
655 PRINT AT 10,3;"Z=";A;B$;B;"I";AT 11,3;"--------"
660 LET U=64: LET V=128
670 LET J=T(1): LET H=T(2)
680 GO SUB 800
690 PLOT 151,128: DRAW 80,0
700 PLOT 191,88: DRAW 0,80
702 LET D$="+": IF D<0 THEN  LET D$=""
705 PRINT AT 10,20;"Z=";C;D$;D;"I";AT 11,20;"--------"
710 LET U=191: LET V=128
720 LET J=T(3): LET H=T(4)
730 GO SUB 800
740 PLOT 71,40: DRAW 80,0
750 PLOT 111,0: DRAW 0,80
752 LET Y$="+": IF Y<0 THEN  LET Y$=""
755 PRINT AT 1[illegible],19;"Z=";X;Y$;Y;"I";AT 14,19;"----------";AT 16,19;"----------"
760 LET U=111: LET V=40
770 LET J=T(5): LET H=T(6)
780 GO SUB 800
```



```
790 PRINT #0;"PULSA CUALQUIER TECLA"
795 PAUSE 0: GO TO 70
800 REM ** DIBUJO DEL VECTOR **
810 PLOT U,V: DRAW J,H
820 RETURN
900 REM ***********************
910 REM **  FACTOR DE ESCALA **
920 REM ***********************
930 DIM T(6)
940 LET MAYOR=-9999
950 LET T(1)=A: LET T(2)=B
960 LET T(3)=C: LET T(4)=D
970 LET T(5)=X: LET T(6)=Y
980 FOR I=1 TO 6
990 IF ABS (T(I))>MAYOR THEN  LET MAYOR=ABS (T(I))
1000 NEXT I
1010 FOR I=1 TO 6
1020 LET T(I)=(T(I)*40)/MAYOR
1030 NEXT I
1040 RETURN
1100 LET RR=INT (R): LET RX=R-RR
1110 LET R$=STR$ RX+"   "
1120 LET R=RR+VAL R$( TO 3)
1130 IF RX=0 THEN  LET R=RR
1140 RETURN
```

Programa 2. Operaciones con números complejos y representación gráfica para el Spectrum.

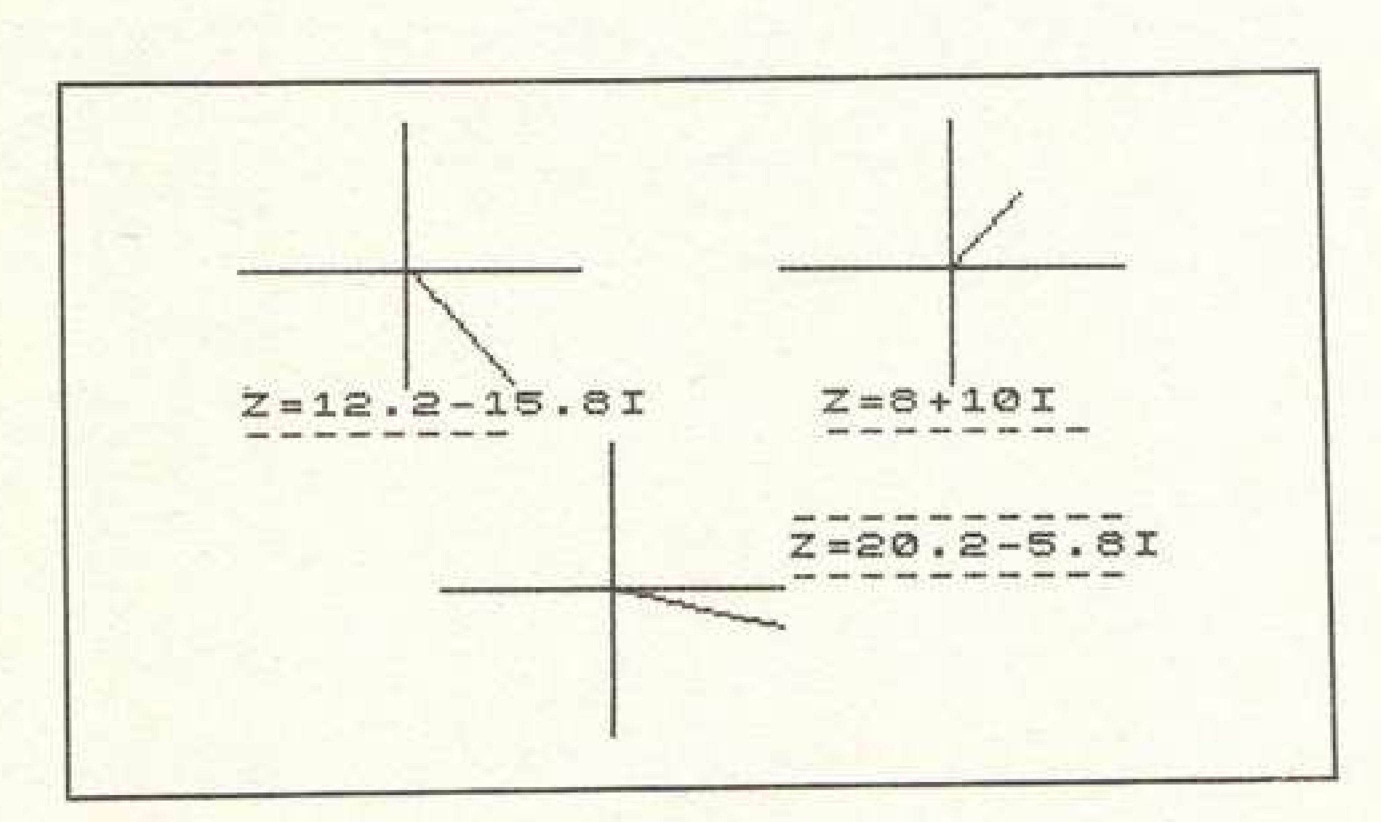


Fig. 4. Representación grafica de los números complejos y del resultado de la operación elegida, en este caso la suma.

Modificaciones para otros equipos

```
5  REM ***************************************
10 REM ***   PROGRAMA DE NUMEROS COMPLEJOS   ***
15 REM ***                                   ***
20 REM *** PARA IBM,AMSTRAD,MSX Y COMMODORE ***
25 REM ***                                   ***
30 REM ***************************************
40 CLS
50 INPUT "PRIMER COMPLEJO";A,B
60 INPUT "SEGUNDO COMPLEJO";C,D
70 CLS
80 PRINT "1.SUMA"
90 PRINT "2.RESTA"
100 PRINT "3.PRODUCTO"
110 PRINT "4.DIVISION"
120 PRINT:PRINT
130 INPUT "ELIGE OPCION";N
140 CLS
150 ON N GOSUB 500,1000,1500,2000
160 REM ********************
170 REM **** RESULTADOS ****
180 REM ********************
190 PRINT "PRIMER COMPLEJO:";"(";A;",";B;")"
200 PRINT "SEGUNDO COMPLEJO:";"(";C;",";D;")"
210 PRINT "RESULTADO:";"(";X;",";Y;")"
220 PRINT:PRINT
230 INPUT "OTRA OPERACION (S/N)";X$
240 X$=LEFT$(X$,1)
250 IF X$="S" OR X$="s" THEN GOTO 70
260 END
500 REM ********************
510 REM ****   SUMAS    ****
520 REM ********************
530 X=A+C
540 Y=B+D
550 PRINT "SUMA DE DOS COMPLEJOS"
560 RETURN
1000 REM ******************
1010 REM ****  RESTA  ****
1020 REM ******************
1030 X=A-C
1040 Y=B-D
1050 PRINT "RESTA DE DOS COMPLEJOS"
1060 RETURN
1500 REM ***********************
1510 REM ****   PRODUCTO    ****
1520 REM ***********************
1530 X=A*C-B*D
1540 Y=A*D+B*C
1550 PRINT "PRODUCTO DE DOS COMPLEJOS"
1560 RETURN
2000 REM *************************
2010 REM ****    DIVISION     ****
2020 REM *************************
2030 X=(A*C+B*D)/(C*C+D*D)
2040 Y=(B*C-A*D)/(C*C+D*D)
2050 PRINT "COCIENTE DE DOS COMPLEJOS"
2060 RETURN
```

Versión estándar del programa.

MODIFICACIONES PARA COMMODORE

```
40 PRINT CHR$(147)
70 PRINT CHR$(147)
140 PRINT CHR$(147)
250 IF X$="S" THEN GOTO 40
```

SOCIEDAD

Test de Prehistoria

Con el programa que a continuación presentamos se inicia el capítulo que dedicamos al estudio de la Historia.

Lógicamente, y atendiendo a su orden cronológico, comenzamos por la Prehistoria.

— La función social que presenta la Prehistoria da un ámbito universal que evita los particularismos actuales y fomenta la solidaridad humana, es decir, identifica el pasado común de todos los pueblos en un mismo hecho: la aparición del hombre sobre la Tierra.

En cuanto a su estudio, nos damos cuenta de que es uno de los más atrayentes, sobre todo porque no necesita de ningún tipo de formación literaria, sino que más bien apela a la imaginación histórica.

— Su función educativa es otra de las más importantes que cabe destacar.

Como ya hemos mencionado, desarrolla la imaginación histórica y nos libera de su contexto limitado.

Nos coloca frente a las fronteras del conocimiento y nos enseña a utilizar los recursos de la técnica moderna.

Estimula el interés por la geografía.

Y, por último, nos muestra la historia de la estética de la humanidad.

La Prehistoria es una ciencia humana. Trata del hombre, no de la Naturaleza, es parte de la historia humana y aunque está cerca de la Ciencia Natural no la trata, sino que se sirve de ella.

Podemos decir que su interés es el de cualquier actividad que además de entretener y educar constituye un beneficio para la sociedad en general, ya que contribuye a la integración social y a la solidaridad.



```
10 REM *************************
20 REM ** TEST DE PREHISTORIA **
30 REM *************************
40 CLS
50 DIM A$(25):DIM B$(75)
60 DIM C(25):DIM D(10)
70 PRINT
80 PRINT "           ##### #   # ##### #     # ##### #   #"
90 PRINT "           #      #  #  #  # # # # # #    # #  #"
100 PRINT "           ###    ##   ##### #  #  # ###  #  # #"
110 PRINT "           #      #  #  #   # #    # #     #   ##"
120 PRINT "           ##### #    # #   # #    # ##### #   #"
130 PRINT
140 PRINT
150 PRINT "    **** **** **** *   * * **** ***** **** **** * ****"
160 PRINT "    * *  *  * *    *   * * *      *  *  * *  * * *  *"
170 PRINT "    **** ***  ***  ***** * ****   *  *  * ***  * ****"
180 PRINT "    *    *  * *    *   * *    *   *  *  * *  * * *  *"
190 PRINT "    *    *  * **** *   * * ****   *  **** *  * * *  *"
200 PRINT
210 PRINT
220 REM ***************************************
230 REM ** LECTURA DE PREGUNTAS Y RESPUESTAS **
240 REM ***************************************
250 FOR N=1 TO 25
260 READ A$(N)
270 NEXT N
280 FOR N=1 TO 75
290 READ B$(N)
300 NEXT N
310 FOR N=1 TO 25
320 READ C(N)
330 NEXT N
340 FOR N=1 TO 10
350 LET D(N)=0
360 NEXT N
370 PRINT"           EL EXAMEN CONSTA DE DIEZ PREGUNTAS"
380 PRINT"                      TIPO TEST."
390 PRINT
400 PRINT"   DEBERAS CONTESTAR EL NUMERO DE LA RESPUESTA CORRECTA"
410 PRINT
420 FOR Q=1 TO 3000:NEXT Q
430 LET X=0: LET A=0: LET B=0
440 REM *******************************************
450 REM ** GENERACION DE 10 PREGUNTAS ALEATORIAS **
460 REM *******************************************
470 RANDOMIZE TIME
480 LET W=INT(RND*25)+1
```

```
490 CLS
500 FOR N=1 TO 10
510 IF D(N)=W THEN GOTO 470
520 NEXT N
530 LET X=X+1
540 LET D(X)=W
550 PRINT A$(W)
560 PRINT
570 PRINT
580 PRINT
590 PRINT TAB(2):B$((3*W)-2)
600 PRINT
610 PRINT TAB(2):B$((3*W)-1)
620 PRINT
630 PRINT TAB(2):B$(3*W)
640 PRINT: PRINT
650 REM ***************
660 REM ** RESPUESTAS **
670 REM ***************
680 INPUT "RESPUESTA: ";C
690 IF C=C(W) THEN GOSUB 740
700 IF C<>C(W) THEN GOSUB 820
710 CLS
720 IF X=10 THEN GOTO 900
730 GOTO 470
740 REM ***********************
750 REM ** RESPUESTA CORRECTA **
760 REM ***********************
770 PRINT
780 PRINT"                     CORRECTO"
790 FOR Q=1 TO 1000: NEXT Q
800 LET A=A+1
810 RETURN
820 REM *************************
830 REM ** RESPUESTA INCORRECTA **
840 REM *************************
850 PRINT
860 PRINT"                     INCORRECTO"
870 FOR Q=1 TO 500: NEXT Q
880 LET B=B+1
890 RETURN
900 REM **********************
910 REM ** RESULTADOS EXAMEN **
920 REM **********************
930 PRINT TAB(5):"   RESULTADOS DEL EXAMEN"
940 PRINT TAB(5):"____________________________":PRINT :PRINT
950 PRINT TAB(5):"   RESPUESTAS CORRECTAS   ":A:PRINT :PRINT
960 PRINT TAB(5):"   RESPUESTAS INCORRECTAS   ":B:PRINT:PRINT
970 IF A>=5 THEN GOTO 1020
980 PRINT TAB(5):"****************************"
990 PRINT TAB(5):"**   DEBES ESTUDIAR MAS   **"
1000 PRINT TAB(5):"****************************"
1010 GOTO 1040
1020 IF A>8 THEN PRINT TAB(5):"   ERES UN BUEN ESTUDIANTE.   ¡SOBRESALIENTE!":
     GOTO 1040
1030 PRINT TAB(5):"   HAS APROBADO"
1040 INPUT "¿QUIERES HACER OTRO EXAMEN?":Z$
1050 IF Z$="S" OR Z$="s" THEN GOTO 340
1060 END
1070 DATA"¿QUE FENOMENO DELIMITA EL FINAL DE LA PREHISTORIA?","LAS DIVISIONES DE
 LA PREHISTORIA ATIENDEN A:"
1080 DATA"¿CUAL ES LA PRINCIPAL DIFERENCIA ENTRE EL HOMBRE DEL PALEOLITICO Y LOS
 ANIMALES?","¿A QUE ESPECIE PERTENECE EL HOMBRE ACTUAL?","LA APARICION DE UTENSI
LIOS DE CAZA Y DE PESCA SE DA EN:"
1090 DATA"¿QUE FUNCIONES PRIMORDIALES TENIAN LAS CUEVAS?","¿QUE SENTIDO TENIAN L
AS PINTURAS RUPESTRES?","¿COMO SE LLAMAN LAS PRIMERAS MUESTRAS ESCULTORICAS?"
1100 DATA"EN EL NEOLITICO LA RELACION HOMBRE-ANIMAL ES:","EN EL NEOLITICO ES NOT
ABLE LA APARICION DE:"
1110 DATA"LA CERAMICA Y LOS TEJIDOS HACEN SU APARICION EN:","EL PRIMER METAL QUE
 SE UTILIZA ES EL:","LAS PRIMERAS ACTIVIDADES ECONOMICAS SE DEBEN A:"
1120 DATA"¿CUAL ES LA RAZA QUE TIENE MAS CAPACIDAD CRANEAL?","¿CUAL ES LA CULTUR
AMAS ANTIGUA QUE EXISTE?","LAS SOCIEDADES AGRARIAS DEL NEOLITICO SE DIFUNDEN HAC
IA:"
1130 DATA"LA CULMINACION DE TODO EL PROCESO NEOLITICO DA LUGAR A:","LOS FOSILES
MAS ANTIGUOS CONOCIDOS SON:","EN QUE ERA COMIENZA EL PALEOLITICO?"
1140 DATA "¿CUANTAS FUERON LAS GLACIACIONES?","¿EN QUE PERIODO APARECE EL HOMO S
APIENS?","EN LA P.IBERICA EL HOMINIDO MAS PARECIDO AL HOMBRE ACTUAL ES:"
1150 DATA "¿EL INSTRUMENTO DE ESTUDIO MAS IMPORTANTE EN LA EVOLUCION DEL HOMBRE
ES:","EL CAMBIO DE ENTERRAMIENTOS COMUNES A INDIVIDUALES SE DEBE A:"
1160 DATA "¿QUE MATERIALES UTILIZABA EL HOMBRE EN EL PALEOLITICO INFERIOR?"
```

```
1170 DATA "1 LA APARICION DEL FUEGO","2 LA INVENCION DE LA ESCRITURA","3 LA APAR
ICION DE LA AGRICULTURA","1 LAS RELACIONES DE LOS HOMBRES","2 LOS LUGARES DON DE
 VIVIAN","3 LOS MATERIALES QUE UTILIZABAN"
1180 DATA "1 LOS LUGARES EN QUE VIVIAN","2 SUS ACTIVIDADES COMUNES","3 LA FABRIC
ACION DE UTILES PROPIOS","1 CROMAGNON","2 SAPIENS","3 NEARDERTHAL"
1190 DATA "1 PALEOLITICO SUPERIOR","2 MESOLITICO","3 PALEOLITICO INFERIOR","1 RE
FUGIO","2 LUGAR DE REUNION","3 CEMENTERIO","1 DECORATIVO","2 MAGICO-RELIGIOSO","
3 LUDICO"
1200 DATA "1 ESTATUILLAS","2 CERAMICAS","3 VENUS","1 MAS LEJANA","2 NO EXISTE","
3 DE DOMINIO","1 CAZA Y PESCA","2 AGRICULTURA Y PESCA","3 GANADERIA Y AGRICULTUR
A","1 PALEOLITICO","2 NEOLITICO","3 EDAD DEL BRONCE"
1210 DATA "1 COBRE","2 HIERRO","3 BRONCE","1 EXCEDENTES DE ALIMENTACION","2 SEDE
NTARISMO","3 UTILIZACION DEL METAL","1 AUSTRALOPHITECUS","2 HOMO ERECTUS","3 NEA
RDERTHAL"
1220 DATA "1 ABBEVILLENSE","2 MUSTERIENSE","3 MAGDALENIENSE","1 EL NOROESTE DE E
UROPA","2 EL SUR DE AFRICA","3 EL OESTE POR EL MEDITERRANEO"
1230 DATA "1 LA REVOLUCION URBANA","2 LA REVOLUCION SOCIAL","3 LA REVOLUCION REL
IGIOSA","1 LOS DE LA P.IBERICA","2 LOS DE AFRICA DEL SUR","3 LOS DE MESOPOTAMIA"
1240 DATA "1 SECUNDARIA","2 PRIMARIA","3 CUATERNARIA","1 TRES","2 DOS","3 CUATRO
","1 PALEOLITICO SUPERIOR","2 NEOLITICO","3 PALEOLITICO INFERIOR"
1250 DATA "1 NEARDERTHAL","2 PITHECANTROPUS","3 CROMAGNON","1 LAS PINTURAS","2 L
OS MONUMENTOS","3 LOS FOSILES","1 APARICION DE LA PROP.PRIVADA","2 ESPECIFICACIO
N DE CLASES SOCIALES","3 DISTINTAS CREENCIAS RELIGIOSAS","1 MADERA, METAL, PIEDR
A"
1260 DATA "2 HUESO, PIEDRA BRONCE","3 MADERA, PIEDRA, HUESO"
1270 DATA 2,3,3,2,1,1,2,3,3,3,2,1,1,1,1,2,1,2,3,3,1,3,3,1,3
```

Programa 3. Test de Prehistoria.

El programa es válido para el IBM-PC. En los demás equipos habría que efectuar las siguientes modificaciones:

SPECTRUM

```
50 DIM A$(25,80):DIM B$(75,40)
420 PAUSE 0
790 PAUSE 100
870 PAUSE 100
1060 GOTO 9999
```

Además, habría que eliminar la línea 470.

COMMODORE

```
40 PRINT CHR$(147)
480 LET W=INT(RND(1)*25)+1
490 PRINT CHR$(147)
710 PRINT CHR$(147)
```

Además, habría que eliminar la línea 470

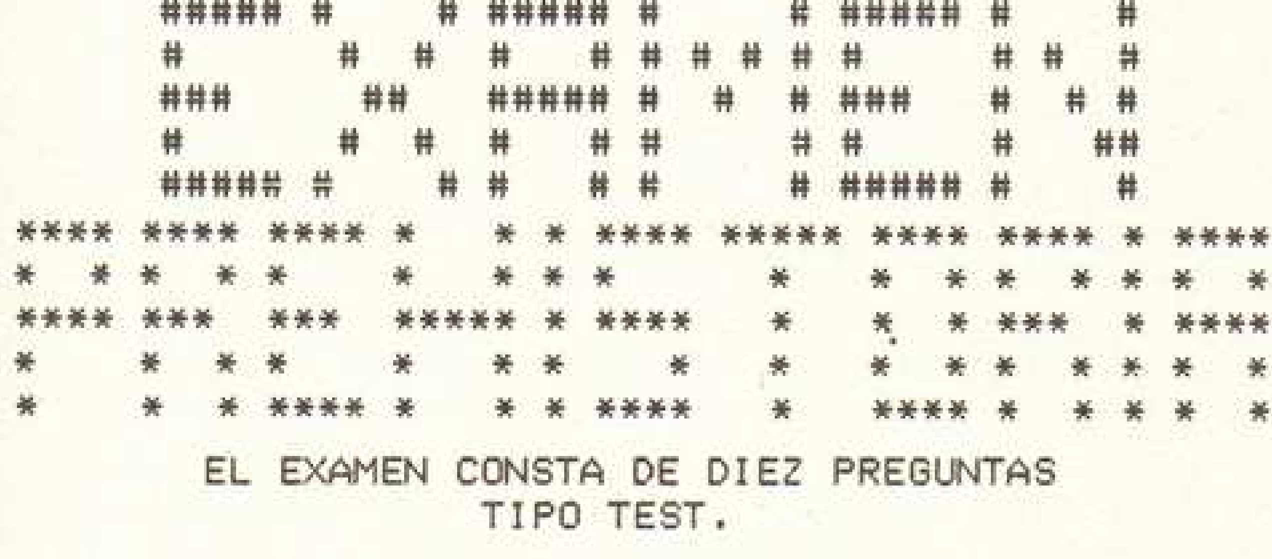
MSX

```
480 LET W=INT(RND(1)*25)+1
```

Además, habría que eliminar la línea 470

AMSTRAD

Habría que eliminar la línea 470




```
DEBERAS CONTESTAR EL NUMERO DE LA RESPUESTA CORRECTA

¿QUE SENTIDO TENIAN LAS PINTURAS RUPESTRES?

 1 DECORATIVO

 2 MAGICO-RELIGIOSO

 3 LUDICO

RESPUESTA: ?
```

Fig. 5. Ejemplo de ejecución del programa de Prehistoria.

El programa es un pequeño examen consistente en diez preguntas tipo test sobre temas básicos de Prehistoria.

Este programa tiene una pequeña "base de datos" formada por veinticinco preguntas y sus tres correspondientes respuestas a cada una.

Aleatoriamente, en cada ejecución del programa se eligen diez de estas veinticinco preguntas y se presentan sucesivamente al usuario para que opte por una de las tres posibles respuestas.

Indagando más en la construcción del programa vemos dos matrices C y D.

El uso de la primera matriz es el de almacenamiento del número de las respuestas correctas.

En D se van almacenando las preguntas que se van formulando para que no se repitan en la ejecución del programa.

Por último hay que hacer resaltar que las preguntas y respuestas pueden ser renovadas sin más que cambiarlas en los respectivos DATAs, teniendo la precaución de colocar las preguntas y respuestas en el mismo orden en dichas sentencias DATA.

PARA LOS MAS JOVENES

Tablas de multiplicar

Este programa te ayudará a dos cosas: te servirá para aprender las tablas de multiplicar y además te será útil para prácticar y probar tu habilidad si ya sabías.

Si crees que no tienes problemas y que ya sabes multiplicar podrías empezar eligiendo la opción TEST. Si realmente has aprendido bien y no cometes errores ¡ENHORABUENA!

Si te equivocas pulsa la opción tablas y podrás repasar y ver cuáles eran tus errores. Cuando creas que ya te lo sabes bien, vuelve a la opción TEST hasta que no cometas ningún error. Puedes hacer competiciones con tus amigos y ver quién juega más veces sin equivocarse.

```
10 REM *****    MULTIPLICACION    *****
20 CLS: PRINT "*****************************"
30 PRINT "**  TABLAS DE MULTIPLICAR  **"
40 PRINT "*************************"
50 FOR x=1 TO 500:NEXT x: CLS
60 PRINT "     ELIGE LA OPCION DESEADA."
65 PRINT: PRINT: PRINT: PRINT
70 PRINT "                  PULSA": PRINT
71 PRINT "                        F=FINALIZAR"
72 PRINT "                        E=TEST"
73 PRINT "                        T=TABLA"
80 INPUT X$
90 IF X$="F" OR X$="f" THEN GOTO 9999
100 IF X$="E" OR X$="e" THEN GOTO 160
110 IF X$="T" OR X$="t" THEN GOTO 500
120 REM *****************************
130 REM ****   EMPIEZA EL TEST   ****
140 REM *****************************
160 CLS: LET A=INT (RND*10)+1
170 LET B=INT (RND*10)+1
180 PRINT "CUANTAS SON ";a;" * ";b:INPUT c$
200 IF VAL (c$)=a*b THEN PRINT "CORRECTO.":FOR x=1 TO 200:NEXT x:CLS: GOTO 210
205 PRINT "ERRONEO"
210 PRINT "  QUIERES INTENTARLO DE NUEVO.S/N"
230 INPUT S$:CLS
240 IF s$="S" OR s$="s" THEN GOTO 160
250 IF s$="N" OR s$="n" THEN GOTO 60
260 CLS: GOTO 110
500 REM ********************
510 REM **     TABLAS     **
520 REM ********************
530 CLS
540 INPUT "QUE TABLA QUIERES (1-10) ? ",N
550 PRINT "     TABLA DEL  ";N
560 PRINT: PRINT: PRINT
590 FOR I=1 TO 10
600 PRINT "        ";N;" * ";I;" = ";N*I
610 NEXT I
620 PRINT: PRINT: PRINT: PRINT
630 PRINT "PULSA UNA TECLA PARA CONTINUAR"
640 IF INKEY$="" THEN GOTO 640
650 CLS: GOTO 60
```

Programa 4. Programa de multiplicación para el Spectrum.

El programa es válido para el Spectrum. En los demás equipos habría que efectuar las siguientes modificaciones:

IBM

```
90 IF X$="F" OR X$="f" THEN END
150 RANDOMIZE TIME
```

COMMODORE

```
20 PRINT CHR$(147):PRINT "*****************************"
90 IF X$="F" THEN END
```

```
160 PRINT CHR$(147):LET A=INT(RND(1)*10)+1
170 LET B=INT(RND(1)*10)+1
200 IF VAL(C$)=A*B THEN PRINT "CORRECTO":FOR X=1 TO 200:NEXT X:
PRINT CHR$(147):GOTO 210
260 PRINT CHR$(147)
530 PRINT CHR$(147)
640 GET A$:IF A$="" THEN GOTO 640
650 PRINT CHR$(147):GOTO 60
```

MSX

```
90 IF X$="F" OR X$="f" THEN END
160 CLS:LET A=INT(RND(1)*10)+1
170 LET B=INT(RND(1)*10)+1
```

AMSTRAD

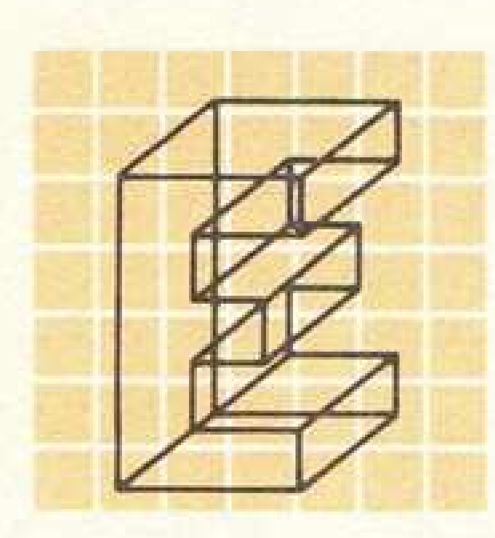
```
90 IF X$="F" OR X$="f" THEN END
```

```
TABLA DEL   7

 7 * 1 = 7
 7 * 2 = 14
 7 * 3 = 21
 7 * 4 = 28
 7 * 5 = 35
 7 * 6 = 42
 7 * 7 = 49
 7 * 8 = 56
 7 * 9 = 63
 7 * 10 = 70

PARA CONTINUAR  PULSA UNA TECLA CUALQUIERA
```

Fig. 6. Tabla de multiplicar del 7.



N el año 1947 parecía que se iba llegando al final, y en la mente de todos estaba claro que los ordenadores se modificarían y mejorarían sus características, pero seguirían con la misma arquitectura (por lo general, la de von Neumann). Sin embargo, en ese año de 1947 tuvo lugar un descubrimiento de enorme trascendencia: el transistor. La palabra transistor deriva de las palabras inglesas *transfer, resistor*, que significa resistencia de transferencia. Este componente electrónico surgió como resultado de las necesidades de la compañía Bell Telephone System, que necesitaba mejorar sus comunicaciones; y en especial sus amplificadores. El transistor se fabricó con un pequeño pedazo de material semiconductor (los semiconductores ofrecen una resistencia al paso de la corriente que es intermedia entre conductores y aislantes); implantando en ese material algunas partículas de «impurezas» se consigue que la conductividad del material (conductividad = inverso de la resistencia), varíe con la tensión aplicada a sus patillas.

Desde luego, aunque los primeros transistores no eran muy fiables (además eran ruidosos), eran bastante más seguros que las válvulas. Estas eran mucho mayores en tamaño, necesitaban tensiones superiores y, por tanto, disipaban mucho más calor, creando los consiguientes problemas.

Pero retrocedamos unos años, y veamos lo que aconteció a nuestros presuntos diseñadores del primer ordenador o máquina universal que utilizaba válvulas.

A finales de 1949, ya superado su trabajo en el ENIAC, Ecker: y Mauchly abandonaron la Universidad de Pennsylvania y pensaron en sus intereses personales tanto tiempo abandonados. Formaron su propia compañía y su primer contrato fue para una corporación de aviación, la Northrop Aviation. Su objetivo era diseñar un ordenador verdaderamente seguro. Con esta idea crearon un ordenador con un diseño secuencial, no como los modernos ordenadores, en los que la información se puede transferir de forma simultánea por varias líneas en paralelo. El BINAC, desde luego, era un ordenador seguro. La verdad es que eran dos ordenadores, uno de los cuales estaba «en reserva» para cuando el otro no pudiera funcionar. Además, el sistema de almacenamiento de la información también era novedoso, ya que utilizaba cinta magnética. Muchas de las válvulas también habían sido sustituidas por diodos semiconductores. El si-

¿Sabía usted que...

El lector recordará que hablamos de una serie de ordenadores, los Colossi, desarrollados en Bletchley (Inglaterra) por el profesor Alan Turing.

Pues bien, esas máquinas también tenían una finalidad militar.

Durante la Segunda Guerra Mundial los británicos obtuvieron a través de los polacos una máquina de criptografía llamada Enigma. Era una oportunidad única que no podían desaprovechar, y el Servicio de Inteligencia, naturalmente con otras siglas, encargó al profesor Turing la familia de ordenadores Colossus con el fin de utilizarlos en descifrar los mensajes criptográficos que iban obteniendo. En una pequeña finca rural próxima a Bletchley se instaló un enorme equipo de escuchas muy motivados en desarrollar su labor.

Las máquinas llevaron a cabo magníficamente su cometido, siendo una ayuda considerable en la toma de decisiones trascendentales.

guiente proyecto de la firma tiene una importancia histórica, ya que se trata ni más ni menos que del primer ordenador comercializado, es decir, fabricado en serie y con el objetivo de venderlo a simples empresas civiles, no a organismos estatales o paraestatales. El producto creado es el UNIVAC I. (El nombre UNIVAC es una abreviatura de Universal Automatic Calculator.)

Pero ¿cuándo se popularizaron realmente los ordenadores? Pues no, no fue calculando trayectorias de planetas, ni estudiando la estructura molecular, el ordenador se popularizó cuando pronosticó y acertó en la victoria del presidente Eisenhower en las elecciones americanas de 1952. A partir de ese momento, los distintos canales de televisión norteamericana comenzaron a preocuparse de la naciente informática, creando concursos, etcétera.

Anteriormente, la empresa de Eckert y Mauchly fue absorbida por la Remington Rand, que más tarde se fusionó con Sperry, y en nuestros días, fusionada con otras compañías (Honeywell), se llama UNISYS.

En 1952 Schockley publica unos trabajos sobre otro tipo de transistores: los transistores de efecto de campo o FET. En este tipo de transistores la innovación fundamental era que el flujo de corriente podía controlarse variando un campo magnético perpendicular al flujo. De este modo los transistores realizaban un efecto de válvula de control y no eran simplemente «filtros» o «amplificadores». En el año 1953 el laboratorio de la Bell Telephone consigue fabricar un tipo de transistor llamado «unión FET o JFET». G. Dacey y M. Ross fueron los que estudiaron el efecto campo en los transistores, pero sus mejoras resultaron demasiado costosas para la compañía, que no podía utilizarlos en todas las aplicaciones. Hubo que esperar hasta 1960 para obtener un transistor que fuera «bueno y barato». Naturalmente, era de efecto campo, pero el material utilizado era silicio (uno de los materiales más comunes y, por tanto, más baratos de nuestro planeta), y el aislante era dióxido de silicio, también relativamente fácil de obtener, ya que estamos utilizando silicio. Su tamaño era muy pequeño, lo que es otra ventaja adicional. El paso de corriente se controla aplicando una determinada tensión al electrodo de puerta. Podemos decir que los transistores han sido el vehículo natural para llegar a los «chips» o circuitos integrados.

En 1954 la Texas Instruments comienza a fabricar masivamente transistores de silicio, ya que la demanda de este tipo de componentes aumentaba, y en la mayoría de las aplicaciones en las que no se requiriera mucha potencia.

Hemos hablado del transistor, porque supuso una revolución en la electrónica. Este pequeño componente se había creado pensando en que sustituyera a los amplificadores de válvulas. Sin embargo, se utilizó para muchas más cosas. En los ordenadores, por ejemplo, sustituyó a las válvulas de vacío en la mayor parte de los circuitos, mejorando notablemente la velocidad de operación. Su robustez (la simplicidad de su constitución) y su tamaño, los hacían más seguros y baratos que las válvulas. El cambio que supuso en la arquitectura de los ordenadores la aparición de los transistores fue tal que marcó las diferencias entre uno y otro tipo de ordenadores. Así, los primeros (que llevaban válvulas), se les llamó Ordenadores de la Primera Generación, y a los que llevaban transistores, Ordenadores de la Segunda Generación.

El ordenador típico de la década de los años 1948 a 1958 tiene la siguiente configuración externa: un armario grande lleno de condensadores, resistencias, cables, válvulas, etcétera; otro armario algo más pequeño para la entrada de la alimentación; una consola de operación; una unidad para la entrada de los datos (que generalmente consistía en una lectora de fichas perforadas o de cinta de papel), y una unidad de salida (también por tarjetas o cinta perforada).

Pero no se asuste el lector; todo lo anterior constituye sólo los «trastos mínimos», ya que ciertas memorias, por ejemplo, de dos niveles que llevan líneas de retardo de mercurio, necesitan colocarse en un receptáculo separado, ya que es fundamental el control de la temperatura. Las tarjetas de entrada de datos se leen a una velocidad de 200 tarjetas por minuto, pero, no nos impresionemos, las de salida se leen y perforan a la mitad de la velocidad anterior. La lectora y la perforadora de cinta de papel suelen colocarse sobre la consola del ordenador, ya que resultan mucho más pequeñas que la de fichas perforadas. En cualquier caso, sólo los empresarios potentes (Lyons, de Inglaterra, etc.) y la suma de pequeños contribuyentes, podía permitirse todavía el enorme coste de las máquinas.

El resultado obtenido con nuestra mo-

¿Sabía usted que...

¿Sabía que uno de los teletextos que funcionan en el Reino Unido, el Prestel, tiene un volumen de doscientas mil páginas?

Este enorme «semanario electrónico» es difundido por línea telefónica. Evidentemente, la comunicación con este libro de nunca acabar es más cara que una comunicación telefónica normal y, además, el precio de la llamada se descuenta automáticamente de la cuenta bancaria del usuario.

Las páginas pueden ser de hasta siete colores diferentes. Tienen una anchura de sesenta caracteres, mayúsculas, minúsculas, cifras y otros signos especiales.

La información del teletexto la proporcionan quinientas entidades distintas. Entre la información a la que puede acceder el usuario se encuentra un mercado de ofertas de trabajo, la guía de ferrocarriles completa, información bursátil, financiera, del tiempo, un diario de noticias, previsiones meteorológicas.

Pero, por si el usuario se desanima, y le parece que ya domina todos los conocimientos anteriores, el Prestel también le ofrece recetas de cocina, anuncios de todo tipo, y, cómo no, novelas tipo serial televisivo.

Una de las enormes ventajas que presenta es que siempre está actualizado, ya que, como puede comprender el lector, esa tarea es sencilla de realizar con un ordenador, e impensable con manuales, libros, etc., de ese volumen.

dernísima máquina es un conjunto de fichas perforadas o una cinta, que son legibles por personas expertas, pero no por cualquier persona ajena a la Informática. Por tanto, para crear un informe presentable con la información procesada es necesario reconvertirla en caracteres legibles por parte de cualquier persona. Para ello deberán cargarse las fichas en el alimentador de la impresora o la cinta en una teleimpresora. Tanto la una como la otra son dispositivos muy lentos, especialmente la teleimpresora, que va escribiendo carácter a carácter, en lugar de hacerlo línea a línea, como lo hace la impresora.

En la década de los sesenta los ordenadores han sufrido algunos cambios importantes. El armario principal se ha dividido en varios armarios (pueden llegar a ser hasta quince). Además se mima mucho más que antes al ordenador: se instala en habitaciones especiales, aisladas de ruidos, con aire acondicionado y una determinada humedad ambiente. La razón de haber desglosado el ordenador en tantos elementos se debe a que al organizarlo por módulos se puede atender más a la demanda de los clientes y venderles una máquina a su medida (al menos eso es lo que nos contaban los vendedores de las casas fabricantes). Estos nuevos ordenadores disponen de una unidad central, a la que se pueden conectar cualquier número y tipo de unidades de entrada y de salida, además de la memoria que se desee. Por tanto, un equipo típico dispondrá de unidades de memoria de alta velocidad (4.000 a 8.000 palabras de 32 a 48 bits), varios tambores magnéticos, con una capacidad de 8.000 a 32.000 palabras, varias unidades de cinta magnética, en las que se puede leer y escribir información a gran velocidad (100.000 caracteres por segundo), lectores de tarjetas (ahora ya son capaces de leer de 400 a 800 fichas por minuto), perforadoras, impresoras que actúan por líneas completas (1.200 líneas por minuto con 100 caracteres por línea, y conectadas directamente a la unidad central). También pueden disponer de lectoras ópticas, de cinta de papel, muy rápidas, y perforadoras de cinta de papel, cuyas velocidades van de 100 a 300 caracteres por segundo. Como puede verse, la técnica ha mejorado en todos los campos: sistemas de almacenamiento, lectoras ópticas, mejores sistemas de grabación, etcétera.

Hemos hablado hasta ahora de ordenadores de la primera y segunda generaciones. Recordemos que los de la primera se caracterizaban por utilizar válvulas, mientras que los de la segunda utilizaban transistores. Enorme innovación, pero ¿se darían por satisfechos los fabricantes, o seguirían invirtiendo enormes sumas de dinero, y los cerebros más eminentes en sus laboratorios de desarrollo? Efectivamente, las empresas fabricantes no se contentaban con los progresos alcanzados, e investigaban para mejorarlos.

Además, la competencia era ya feroz. IBM, que había perdido su oportunidad de introducirse en la fabricación de ordenadores en sus primeros tiempos (no participó en la aparición del UNIVAC, primer ordenador «asequible»), decidió cambiar de rumbo y pasar a fabricar máquinas universales: es decir, ordenadores. Tom Watson hijo fue el artífice de este cambio, oponiéndose al grupo conser-

vador de la empresa, cuya cabeza era su padre, el fundador del imperio. IBM era una empresa muy potente, que fabricaba innumerables modelos de máquinas de oficina, es decir, máquinas que realizaban únicamente ciertos cálculos para los que habían sido programadas. El servicio que ofrecía la empresa era inmejorable, y el viejo señor Watson estaba empeñado en proteger y consolidar su imperio. ¡Le había costado tanto trabajo levantarlo! Pero volvamos, pues, a los laboratorios de investigación. La tercera generación de ordenadores nacía con los circuitos integrados. Como ya sabe el lector, un circuito integrado es una pequeña pieza de material semiconductor en la que se podían incluir varios transistores (hoy en día son miles).

Los ordenadores construidos con esta nueva tecnología tenían grandes ventajas sobre sus antecesores de la segunda generación. Eran más seguros, más rápidos y más baratos. Pensará el lector que estas nuevas tecnologías habían surgido sin el apoyo que ofrecen el Estado, sobre todo en tiempo de guerra.

Efectivamente, la enorme competencia y el espíritu pionero americano crearon muchas empresas, y algunas de ellas comenzaron sus investigaciones en garajes y trasteros. Pero además de todo este enorme esfuerzo, el estado norteamericano comenzó a gastar sumas ingentes en investigación espacial. Y los cohetes necesitaban ordenadores potentes, pero cuyo peso fuera el menor posible. Por esta razón, favorecieron la investigación dirigida hacia este tipo de circuitos integrados.

En los primeros proyectos de la NASA los ordenadores permanecían en tierra, y las órdenes eran enviadas desde los laboratorios por sistemas complejos de comunicaciones. Hoy en día, en el Kennedy Space Center, de Cabo Cañaveral, y en el Johnson Space Center, de Houston, existen potentes ordenadores que colaboran en los distintos proyectos, pero el papel principal ha pasado a los ordenadores de a bordo, que controlan motores, instrumentos de vuelo y sistemas de puesta en órbita y regreso a la Tierra, e incluso el cabeceo que pudiera marear a los tripulantes.

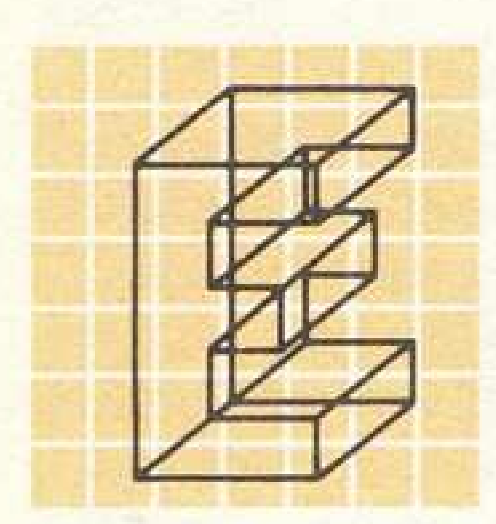
¿Sabía usted que...

¿Sabía usted que las fichas perforadas suponían unos ingresos nada despreciables para la IBM?

Desde 1930 a 1950 la venta de fichas perforadas supuso para la IBM, aproximadamente, un 25% de sus ingresos. Es más, se piensa que esa estimación es mucho más baja que la real, sobre todo en la década de 1930 a 1940.

Una de las razones de estas enormes ganancias se debía a que las máquinas IBM sólo podían utilizar fichas IBM. Esta política supuso para IBM un control y unos beneficios enormes, aunque finalmente le supondría pleitos, acusada de monopolio. Los norteamericanos siempre han sido exquisitos defensores de la libre competencia, y la potente empresa era demasiado próspera para que otras empresas le dejaran el campo libre.

Seguridad e informática (I)



L problema de la seguridad en Informática es clave, porque confluyen en este tema dos fenómenos diferentes, pero coadyuvantes: por un lado, la falta de seguridad general que se va extendiendo en todo el mundo, con los desarrollos tecnológicos (problemas de terrorismo, problemas de espionaje industrial, extorsión, etc.); por otro lado, los avances en Informática hacen que los equipos implicados sean de gran valor, y sobre todo, que las informaciones y datos almacenados, así como las actividades desarrolladas sean día a día más importantes: en efecto, antiguamente, si se estropeaba —fortuita o voluntariamente— un ordenador, se pasaba a otro, y si se perdía un archivo, se tomaba la copia y se seguía trabajando con ella; hoy en día, no sólo la recuperación de información es más complicada, sino que los procesos en tiempo real exigen aumentar enormemente las protecciones que se adopten para tener la seguridad de que no se producirán fallos.

Además, los avances de la electrónica y las comunicaciones por un lado, y de la informática por el otro, ponen en manos del posible defraudador, terrorista... o simple curioso, "herramientas" hace unos años impensables. De hecho, las violaciones de los sistemas informáticos son cada día mayores y, en concreto los fraudes en entidades financieras están creciendo de manera exponencial desde hace tres o cuatro años en Estados Unidos; esto se debe a varias razones: por un lado, los sistemas bancarios son más abiertos y públicos que los que utilizan, por ejemplo, las empresas en sus laboratorios de desarrollo, o el Ejército o la NASA; por otro lado, el interés de "saltar" las protecciones en estos sistemas es mayor (y nunca mejor dicho lo del "interés", puesto que aquí el "beneficio" se cuenta en dólares —o millones de dólares—) y el número de personas interesadas en este tipo de fraudes (número de posibles cabezas pensantes defraudadoras) es mucho mayor; además, con la microinformática y los sistemas caseros se dispone, hoy en día, por un precio módico de un equipo informático con una capacidad de proceso enorme. Para comprobar las dificultades existentes en la represión de este tipo de delitos, se puede pensar que, además, es usual que sólo se detecten los fraudes por azar y, por otro lado, la falta de legislación adecuada y el hecho de que los delitos sean sin violencia y no sobre personas (sino contra entidades) hacen que los jueces sean bastante benevolentes (de hecho, en un proceso contra un defraudador de la Pacific Telephone en el año 1968, que había robado un millón de dólares —unos 135 millones de pesetas de hoy— fue condenado a 60 días en una granja penitenciaria y sólo cumplió —por buena conducta— 40 días: hoy tiene un lucrativo negocio como asesor informático dedicado al diseño de sistemas de protección contra fraudes).

Delitos en los sistemas de TEF

Existen tres formas básicas de "delito con el ordenador" respecto de las entidades financieras.

a) Por un lado, se han producido numerosas estafas mediante el desvío fraudulento de fondos (en un sistema general de Transferencia Electrónica de Fondos) a una cuenta incontrolada; en este caso, se suele detectar la falta del dinero de un modo casi inmediato,

PROTECCION DE LA INFORMACION EN INFORMATICA

A) EN EL PROCESAMIENTO DE LOS DATOS
A.1. CONTRA ERRORES
- Control de procesamiento del archivo adecuado (nombre de archivo, n.° de generación, n.° de orden...).
- Control lógico de la información (a nivel campo) (límites para las variables, campos alfabéticos y/o numéricos...).
- Control lógico a nivel de grupo de datos (controles de total, controles cruzados).
- Control físico de integridad de los datos (bit de paridad, control de operaciones prohibidas, de acceso a áreas reservadas de memoria...).

A.2. CONTRA ACCESOS INCONTROLADOS
A.2.1. Sistemas lógicos
- Palabras—clave generales o parciales.
- Mensajes de interrupción (Mensaje dejado por el operador para detenciones momentáneas del proceso —paradas para desayuno u otras—...).

A.2.2. Sistemas de protección física
- Protecciones físicas de acceso a la sala del ordenador o la sala de terminales. Llaves de control.
- Protección de los archivos magnéticos: cintas y discos (cámaras de seguridad y otros).

A.2.3. Sistemas mixtos
- Sistemas de clave + dispositivo de acceso (terminales físicos de creación de claves en conjunción con rutinas software del ordenador, dispositivos que emiten palabras clave...).
- Sistemas de identificación (reconomiento de la voz, de huellas, reconocedor de la retina...).

B. EN LA TRANSMISION DE LOS DATOS Y TELEPROCESAMIENTO
B.1. EN CUANTO A ERRORES DE TRANSMISION
- Control de integridad de la transmisión (bit de paridad, códigos de detección y/o recuperación de errores...)
- De validez de la transacción (individual) (reconocimiento del mensaje -acknowledgment-, control de n.° de transacción incluido por el operador o añadido automáticamente por el equipo.)

B.2. EN CUANTO A SEGURIDAD DE LOS DATOS
- Ocultación de mensajes (técnicas criptográficas y de enmascaramiento).
- Claves de acceso a la red (palabras simples o a varios niveles, de acceso general o aparte de los datos y/o procesos).
- Dispositivos de seguridad de los terminales (llaves físicas de apertura, interfaces con claves o tarjetas de control...).

pero no se sabe a "dónde ha ido a parar"; el estafador saca posteriormente de su cuenta el dinero, normalmente a través de una cuenta convencional, y en otro punto muy alejado de la oficina donde se produjo la "desviación".

b) Más usuales han sido los casos de obtención de grandes sumas por "redondeo"; las fracciones de centavo de dólar (cada centavo equivale a 1,30 pesetas, aproximadamente) son "desviadas" a una cuenta especial de acumulación: cada usuario no detecta estas pequeñas pérdidas, pero el interesado obtiene sumas importantes con la acumulación, día tras día, de los "redondeos" de miles de operaciones.

c) Otro tercer tipo de delito que se está extendiendo en Estados Unidos es el "vandalismo informático": mediante ordenadores personales y a través de los sistemas de «telebanco» establecidos, se introducen en los grandes sistemas bancarios y pueden llegar a producir grandes daños (destrucción de procesos o de ficheros de datos). La principal dificultad existente para detectar este tipo de delito es que es complicado prever dónde o cómo se va a producir, porque no tiene finalidad de lucro y el interés de estos "gamberros" es sólo producir un daño lo mayor posible y cuanto más en zonas protegidas del sistema, mejor.

Frente a estos tipos de delito, se han desarrollado numerosos sistemas que, globalmente, se pueden agrupar en tres apartados: sistemas de protección física, sistemas de control lógico de acceso y sistemas mixtos.

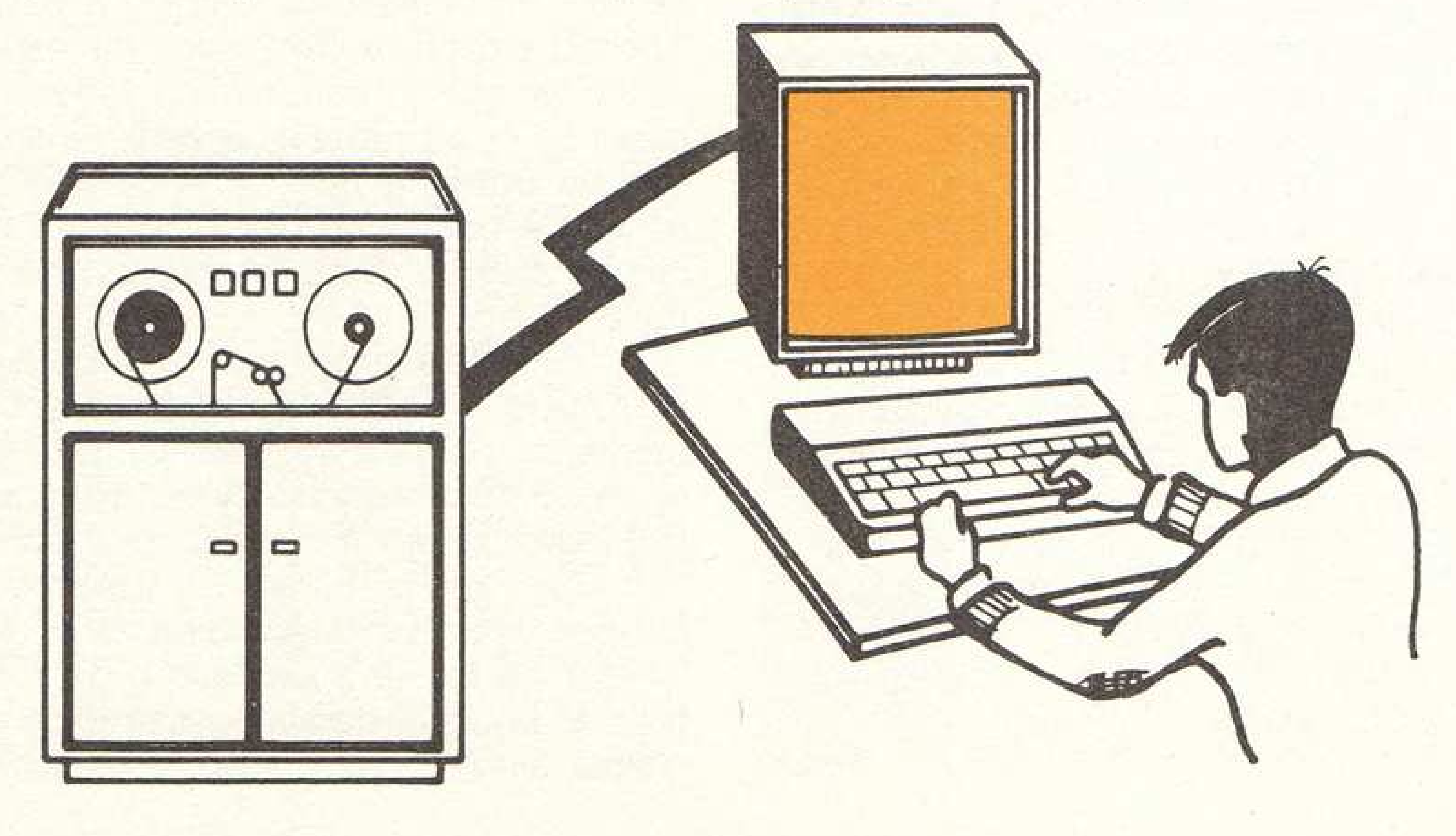
a) Los sistemas de protección física pueden estar instalados bien en el recinto de la sala de los ordenadores o de terminales, bien en las salas de almacenamiento de los soportes magnéticos (cintas y discos). Básica-

mente coinciden con los sistemas generales de seguridad: llaves de puesta en marcha de los equipos, tarjetas magnéticas de acceso, cámaras acorazadas. En los últimos años se han desarrollado enormemente los sistemas de protección biométricos, es decir, que examinan y miden características del cuerpo de la persona que intenta el acceso, para determinar si se le permite pasar o no. Los sistemas biométricos más utilizados son: detector de huella dactilar (se comparan los rasgos básicos de la huella presente —el usuario pone el dedo en un objetivo dispuesto al efecto— con los que el sistema tiene archivados para ese número de clave), geometría y tamaño de la mano (también exige un alto coste y una cantidad de memoria de almacenamiento grande), reconocimiento de la voz (todavía hoy no muy fiable; aún no es comercial), análisis de la firma (es el método más antiguo y más probado; suele ir actualizando las firmas de las personas autorizadas para evitar los problemas de cambio de firma con el paso del tiempo), reconocimiento de retina (semejante al de huellas dactilares, pero más fiable; analiza la disposición de las venillas que surcan la retina), etc.

b) Sistemas de control lógico de acceso son aquellos que permiten el paso o no al usuario en función de que éste facilite la clave adecuada al sistema. El control se realiza por software y básicamente existen dos tipos: aquéllos en que la clave es fija (al menos durante un período de tiempo) para todos los usuarios, o una para cada usuario (caso más normal) y aquellos otros en que la clave varía. De estos últimos, un caso típico es el de "control después del desayuno": es un sistema previsto para que el operador de un terminal pueda cerrar el acceso desde ese terminal provisionalmente cuando se ausenta un breve espacio de tiempo (salida para el desayuno, comida, etc.): el sistema admite un breve mensaje (hasta 50 caracteres) que cierra el acceso y no lo vuelve a abrir hasta que se repita dicho mensaje de control; a veces este sistema se complica con controles adicionales horarios (cierre definitivo si no se reanuda la sesión de trabajo antes de un número prefijado de minutos, control de que la sesión no se reanuda antes de "n" minutos, control de que este cierre "para el desayuno" sólo se hace a ciertas horas del día, etc.).

c) En los sistemas mixtos, se combinan ambas técnicas: por software se controla la introducción de ciertas claves, pero es necesario además alguna operación manual (introducción de llaves o tarjetas de control). Otro grupo de sistemas de seguridad se basa en que el dispositivo de control (normalmente un aparatito parecido a una calculadora) emite una clave especial, o, en presencia de la pantalla del ordenador, genera una clave especial (variable) que es la que el usuario debe introducir, o alguna otra variante sobre este esquema básico.

Pero no es el acceso incontrolado a la información el único peligro que existe en un sistema de proceso de datos. Además, existen causas (fortuitas o no) que pueden producir daños enormes (en ocasiones, irreparables) en la información almacenada o en los programas introducidos. En un sistema informático o de EDP pueden causar mayores trastornos que los defraudadores, el agua o el fuego o cualquier tipo de sabotaje.



El fuego

De los puntos mencionados, probablemente el fuego sea el peligro principal. Cada año se registran numerosos incendios en oficinas. El fuego es un problema crítico en centros con ordenadores por muchas razones: en primer lugar, porque son lugares llenos de material combustible, como papel, cajas, etc. El falso suelo y el cableado que va por él también pueden causar problemas serios.

Desgraciadamente, los sistemas contra incendios no son todo lo modernos que desearíamos, causando casi tanto daño al material como el propio fuego, sobre todo en el caso de las cintas magnéticas. El dióxido de carbono, alternativa actual del agua, resulta también peligroso para los propios empleados, si quedan atrapados en la sala del ordenador.

El fuego es el principal enemigo del ordenador, ya que puede destruir fácilmente ficheros de información y programas, que suelen ser difícilmente sustituibles. El hardware normalmente está asegurado, y el fabricante puede sustituirlo o repararlo generalmente en muy poco tiempo, de forma que los daños físicos del equipo suelen ser el problema menor. La sustitución de las cintas y ficheros es mucho más difícil, y en ocasiones imposible. Las empresas deben tener backups (copias de seguridad) de sus ficheros, y almacenarlas en otros edificios. Esta es una norma de seguridad muy importante, que deberían seguir todas las empresas.

Además de la pérdida de ficheros, y otros daños físicos del equipo, el fuego puede causar otras pérdidas importantes no cubiertas por el seguro: la pérdida del "momento del negocio". La pérdida de semanas o meses puede ser un daño irreparable para cualquier negocio.

El agua

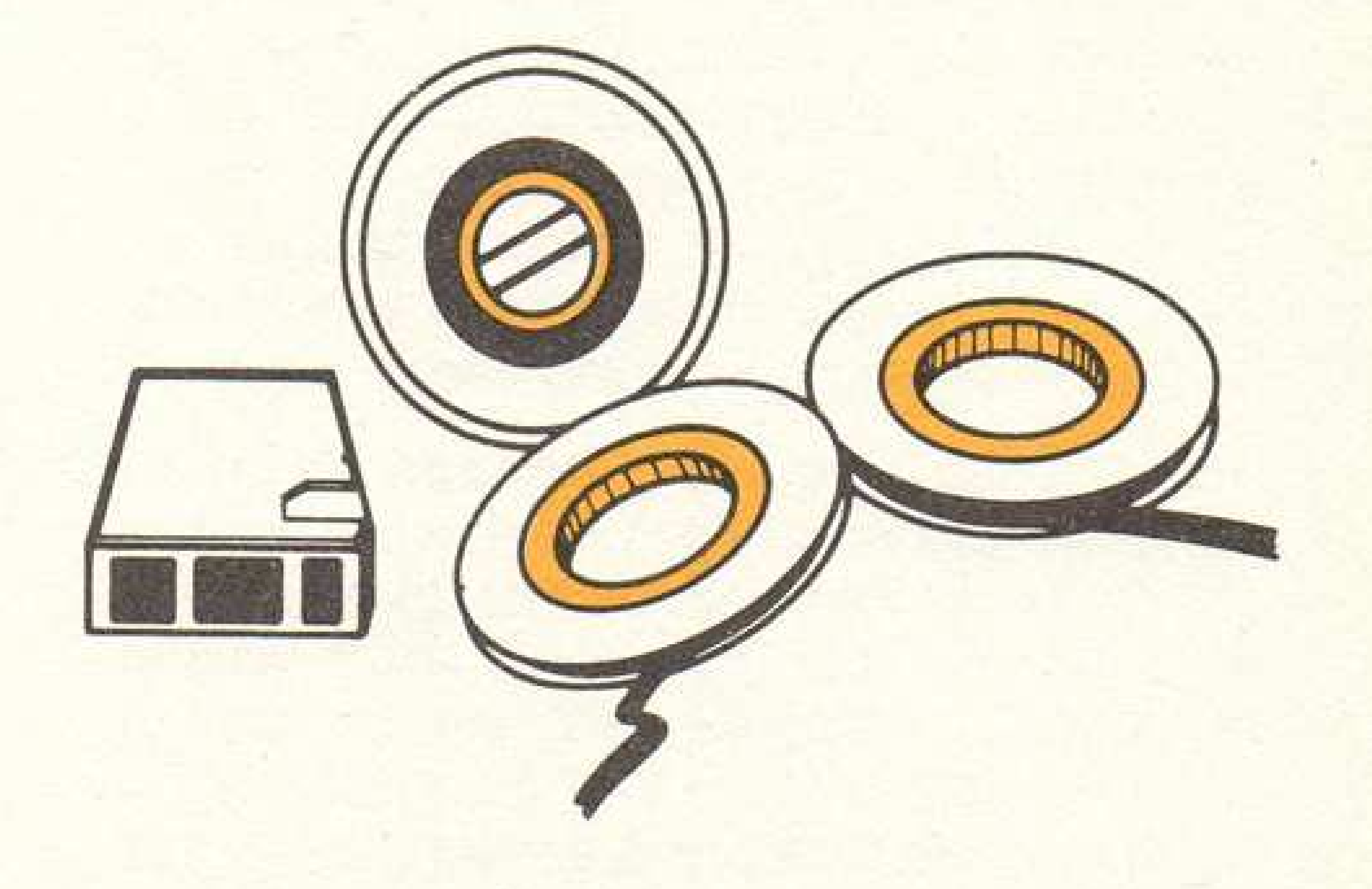
Otro de los agentes importantes de accidentes es el agua. El agua puede entrar en la sala de ordenadores por varios conductos. Los centros de proceso de datos que se encuentran en sótanos, o a nivel de calle, pueden sufrir inundaciones. También pueden reventarse cañerías que vayan por el falso suelo, o en techo o paredes. El agua es una amenaza para cables y en general para el hardware del ordenador, pero no constituye una amenaza seria para las cintas magnéticas. Se ha demostrado que cintas que han estado sumergidas en agua varias horas han podido ser leídas de nuevo (y sin errores) después de un secado de varios días de duración. Sin embargo, el agua puede constituir un peligro serio si a ella se suma el calor. Las cintas magnéticas se destruyen a una temperatura de 54 grados si la humedad es superior al 85%. La temperatura de 54 grados puede parecer muy alta, pero puede alcanzarse, por ejemplo, en un coche cerrado, en verano.

Sabotaje

El peligro más temido en los centros de cálculo de las empresas es el sabotaje. Las empresas que han abordado el problema y han intentado adoptar sistemas de seguridad de alto nivel se han encontrado con un problema arduo de resolver. El saboteador puede ser un empleado de la propia empresa o alguien ajeno a ella.

Los imanes son herramientas muy utilizadas; aunque las cintas estén almacenadas dentro de su funda de protección, una ligera pasada con el imán, y la información desaparece. Con este sistema se puede destruir una habitación llena de cintas en cuestión de minutos. Además, también se pueden introducir partículas de metal, gasolina, suciedad, por el falso suelo, cortar las líneas de comunicación y eléctricas, lanzar bombas, u otros proyectiles, etc.

La seguridad de un centro de cálculo



es otro problema más que debe ser evaluado por la dirección, considerando y calculando los riesgos, las pérdidas y los medios que pueden ponerse en práctica razonablemente para prevenirlos.

Generalmente, una mejora en la seguridad produce importantes beneficios secundarios. Por ejemplo, el cambio de la metodología aplicada a determinadas operaciones conduce con frecuencia a una reducción del índice de errores, a una mejora de la calidad, mejor planificación y a resultados más rápidos.

No existe un plan idóneo o una recomendación sencilla para resolver el problema de la seguridad. No es una situación estática, y requiere un esfuerzo continuo y un estudio periódico de los problemas y soluciones razonables considerando pérdidas, costes, riesgos, etc.

TERMINOLOGIA

GLOSARIO DE TERMINOS UTILIZADOS EN RELACION CON LA SEGURIDAD

ACK (Ver Acknowledge, message).

Acknowledge, message. Reconocimiento de mensaje. Señal de control que envía el receptor de un mensaje para indicar que la transmisión se ha efectuado a satisfacción. Normalmente indica, además, que el receptor está dispuesto para aceptar otro mensaje.

Back-up. Copia de seguridad. Cuando un archivo adquiere un volumen grande (o periódicamente en los procesos usuales) es útil obtener una copia de él, para tener la seguridad de poder reconstruirlo en caso de error o fallo del sistema. Las copias de seguridad se suelen sacar en dispositivos portables (cintas magnéticas o discos removibles en equipos grandes, o disquetes en microordenadores) para poder ser almacenadas en un lugar distinto de donde se encuentra el ordenador.

Biométricos, controles. Controles de seguridad basados en el examen y medición de las características de tamaño, forma y disposición general de ciertos elementos del cuerpo (mano en general, huella dactilar, huella de la retina, etc.).

Bit de paridad. El último bit de un byte de datos (última posición —bit— en un conjunto de ellas —byte—) utilizado para controlar la transmisión o almacenamiento de dicho byte de datos. Si (dependiendo del carácter, letra, número... que se representa) el número de bits que tienen valor 1 es par, el bit de paridad se pone a 1 para que el número total de bits a 1 sea impar; si en el carácter que se representa hay ya un número impar de 1, el bit de paridad se pone a cero. Todo esto en «modo de paridad impar»; la cuenta total debe ser par en «modo de paridad par». Al llegar el carácter a su destino se comprueba si el bit de paridad se adecua a la norma indicada y si no, es que ha habido error en la transmisión de algún(os) bit(s).

Código de detección o recuperación de errores. Procedimiento para detectar si ha habido error en la transmisión de un carácter o grupo de caracteres. Suelen ser uno o varios bits que toman valores de acuerdo con una norma prefijada y que puede ser controlada en destino, para el caso de control de un carácter. El bit de paridad es un bit de detección de errores, aunque limitado lógicamente (si se cambian por error dos bits —de «0» a «1» o de «1» a «0»— la cuenta total de paridad sale bien, aunque haya habido error). Para el control de un grupo de caracteres (o todo un mensaje) se suele usar un carácter completo cuyos bits toman valores en función del contenido del grupo de caracteres cuya transmisión se quiere controlar. Si se añaden controles a un código (más bits o más caracteres de control que los imprescindibles) se puede detectar no sólo que se ha producido un error, sino dónde y, por tanto, corregirlo.

Control de acceso. —Físico. Cuando se impide la entrada al recinto correspondiente al usuario no autorizado, o se impide la conexión física (eléctrica) del equipo o su puesta en marcha. **—Lógico.** Cuando el equipo funciona y el usuario (incluso el no autorizado) puede acceder a él, pero no se

realizan las actividades correspondientes (ni proceso de datos, ni acceso a la información).

Criptografía. Ciencia de la ocultación de mensajes. Técnicas de modificación del texto a transmitir (texto «claro») para que resulte indescifrable (texto «cifrado»); naturalmente, en el punto de destino de la información hay que realizar el proceso inverso para volver a obtener el texto claro.

Check digit. Dígito de control. Carácter o dígito utilizado en el control de la transmisión para la detección (y eventual corrección) de errores. Ver Código de Detección o Recuperación de Errores.

EDP (Electronic Data Processing). Proceso electrónico de datos. Los centros de EDP o Centros de Proceso de Datos (CPD) son los locales donde se ubican los ordenadores electrónicos, los elementos auxiliares de ellos (máquinas y terminales) y los dispositivos de almacenamiento de información.

Generación. El número de generación de un fichero indica una secuencia en las sucesivas versiones que de ese fichero se van produciendo, con las diferentes actualizaciones de datos que se introducen.

Palabras-clave. Código de acceso lógico a un sistema. Las palabras clave (Keyword, en inglés) pueden no sólo permitir o impedir el acceso, sino realizar cierta selección para que cada usuario sólo pueda llegar a los procesos o informaciones para los que está autorizado.

Reconocedor. Se llama así al equipo físico (electrónico y óptico) que analiza las imágenes (huellas dactilares, forma y tamaño de la mano, etc.), y las compara con las de los usuarios autorizados para permitir o no el acceso a un recinto o sistema.

TEF. Transferencia electrónica de fondos. Se llaman así, en general, a los sistemas de comunicación entre ordenadores o entre éstos y ciertos terminales de tipo bancario para hacer las operaciones propias de las instituciones financieras: pagos, cobros, transferencias entre cuentas, etc.

Transacción. Se utiliza para designar un conjunto de datos que han de ser procesados como una unidad.

VOCABULARIO DE INFORMATICA

Background colour. Color de fondo de un gráfico o de una pantalla, en contraposición con el **foreground colour** o color del texto (color con el que se dibujan los gráficos propiamente dichos sobre el fondo **—background—** de la pantalla).

Background, proceso. Ejecución de programas de segundo nivel (prioridad baja), en contraposición a los programas de primer nivel (prioridad alta), o programas del área **foreground,** que son ejecutables en tiempo real.

Backspace, carácter. Configuración ASCII que produce los mismos efectos que la tecla de **backspace** (abreviadamente, **bs**).

Backspace, tecla. Tecla utilizada para retroceder un carácter al escribir un texto en el ordenador, o para que retroceda una posición el carro de la impresora.

Backup. (Ver Copia de seguridad.)

Banco de datos. (Ver Base de datos.)

Banda. A veces se designa con este anglicismo a las cintas magnéticas.

Banda. ancha (estrecha), término utilizado en comunicaciones para distinguir las transmisiones en función del ancho de banda. (Ver Ancho de banda.)

Bandera (flag). Señal o marca que delimita un punto de un conjunto de datos; registro de estado donde se anotan un conjunto de situaciones de interés para el desarrollo de un proceso. Variable lógica que controla el estado de un dispositivo o la característica (verdadera o falsa) de una situación o prueba.

Barra (slash). Carácter de control de numerosas aplicaciones. En operaciones aritméticas, significa dividido por. Como carácter de control se utiliza a veces también la barra inversa.

Barra de Sheffer. Nombre con el que se designa al operador lógico « | », equivalente al NAND (negación del AND = Y lógico).

Barrido. Exploración sistemática de una imagen en la que el haz de electrones analiza de izquierda a derecha y de arriba abajo todos los puntos que la componen.

Base. Valor de referencia especialmente utilizado en los sistemas de numeración como número con cuyas potencias se representan los demás números. (Ver Sistemas de numeración.)

Base. En una potencia, número que debe ser multiplicado por sí mismo tantas veces como indique el exponente.

Base de datos. Conjunto de datos almacenados de forma estructurada y sin redundancias, al que se puede acceder de forma rápida, clasificar, analizar, etc. Normalmente son enormes ficheros de datos que están continuamente actualizándose.

Base de datos distribuida. Base de datos en la que la información está distribuida en varios subsistemas diferentes que generalmente están físicamente separados. Si los subsistemas son relativamente iguales, se

llaman homogéneos; en caso contrario, heterogéneos.

Base de datos relacional. Base de datos almacenados de forma relacionada, es decir, cuya estructura relacional se conforma en el momento en que se establecen las consultas, y no en el momento de su creación.

Batch, proceso en. Proceso por el cual los programas se ejecutan de forma secuencial, sin que medie ninguna acción del operador. En contraposición al proceso en tiempo real, el proceso en **batch** realiza sus tareas en el orden útil sin relación alguna con las actividades externas a dicho proceso.

Baudio. Unidad de medida para la velocidad de las transmisiones de datos. Indica el número de veces que el sistema cambia de estado. En un canal binario, equivale a un bit por segundo. El nombre se debe al científico J. E. Baudot.

BCD. (Ver Notación decimal codificada en binario.)

Beeper. Pequeño altavoz situado en el interior del ordenador que produce un sonido especial.

Benchmark, problema. Problema que sirve para comparar dos o más ordenadores (software y hardware) o un determinado ordenador respecto de las especificaciones impuestas.

Bias. (Ver Polarización.)

Bias. Diferencia entre la media de un conjunto de valores y el valor de referencia. Desviación.

Biblioteca de programas. Conjunto de programas y rutinas de uso común. Los elementos que la componen no necesitan estar relacionados.

Bicondicional. Expresión lógica que combina otras dos expresiones lógicas, de forma que el resultado sólo es verdad si las dos expresiones son ambas verdaderas o ambas falsas.

Bidireccional. Línea de comunicaciones en la que los datos pueden transmitirse en cualquiera de las dos direcciones (en contraposición con **half duplex**).

Biestable. Dispositivo capaz de encontrarse en uno de sus dos estados estables.

Bifurcación. Conjunto de algoritmos que otorgan estructura arbórea a una tabla de decisiones. Una vez estructurada la tabla, se puede ya codificar y crear con ella un programa.

Binaria, búsqueda. Sistema de búsqueda que consiste en ir dividiendo el número de elementos del conjunto en dos partes en cada paso del proceso.

Binaria, notación. Sistema de numeración que utiliza como base 2.

Binario, código. Sistema de codificación que utiliza dos caracteres: 0 y 1.

Binario, dígito. Cualquiera de los caracteres utilizados en la notación binaria, es decir, 0 ó 1. Su abreviación, **bit,** deriva de la expresión inglesa **binary digit.**

Binario, número. Número representado en notación binaria. Por ejemplo, 101 es un número binario equivalente al 5 decimal, y al V romano.

Bit. Unidad elemental de información. Puede ser 1, 0, SI, NO, etc. Representa dos posibles estados. Es una abreviación de la palabra inglesa **binary digit.**

Bit de paridad. Bit que se añade (como sistema de comprobación) a un paquete de información que se va a transmitir. La información está compuesta por otros bits. Por ejemplo, puede hacerse que el bit de paridad sea 1 si el número de bits 1 a transmitir es un número impar, y 0, si es par.

Bit menos significativo. Bit situado en la posición menos significativa, o de menos peso. Generalmente es el bit situado más a la derecha.

Bits por pulgada. Medida de la densidad de los sistemas magnéticos de grabación. Número de bits que pueden grabarse en una pulgada de soporte magnético.